Carnegie Mellon
Software Engineering Institute

Evolutionary Process for Integrating COTS-Based Systems (EPIC)

Building, Fielding, and Supporting Commercial-off-the-Shelf (COTS) Based Solutions

Authors:
Cecilia Albert, Software Engineering Institute
Lisa Brownsword, Software Engineering Institute

In Collaboration with:
Colonel David Bentley, USAF
Thomas Bono, MITRE
Edwin Morris, Software Engineering Institute
Deborah Pruitt, MITRE

November 2002

TECHNICAL REPORT
CMU/SEI-2002-TR-005
ESC-TR-2002-005

Carnegie Mellon
Software Engineering Institute
Pittsburgh, PA 15213-3890

Evolutionary Process for Integrating COTS-Based Systems (EPIC)

Building, Fielding, and Supporting Commercial-off-the-Shelf (COTS) Based Solutions

CMU/SEI-2002-TR-005
ESC-TR-2002-005

Authors:
Cecilia Albert, Software Engineering Institute
Lisa Brownsword, Software Engineering Institute

In Collaboration with:
Colonel David Bentley, USAF
Thomas Bono, MITRE
Edwin Morris, Software Engineering Institute
Deborah Pruitt, MITRE

November 2002

COTS-Based Systems Initiative

This report was prepared for the

SEI Joint Program Office
HQ ESC/DIB
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

Jay Minsky
Contracting Officer's Representative

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

This work was created in the performance of Federal Government Contract Number F19628-00-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 52.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Table of Contents

List of Figures

Acknowledgements

Colonel David P. Bentley, Director, Business Information System Program Office, Electronic Systems Command (ESC/MM), United States Air Force provided the vision for this project. He also provided some of the funding, and much of the impetus that kept the project moving. Thomas L. Bono and Debora M. Pruitt of MITRE Corporation provided a sounding board for the major concepts, and invaluable editorial support. Edwin Morris of the Software Engineering Institute (SEI) clarified our complex thoughts and developed much of the material on COTS component evaluation.

John Foreman and Thomas Brandt supported this project as it became more complex. David Carney, Patricia Oberndorf, and Patrick Place provided valuable insights into the unique aspects of a process framework for COTS-based systems and comments that improved the quality of this document. Fred Long, Marc Kellner, and Fred Hansen of the SEI, and Judy Clapp, Mike Bloom, Pam Martin, and Dave Hart of MITRE Corporation read versions of this document. Their comments made this document much stronger.

Abstract

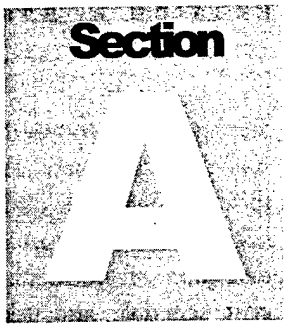
Government and private organizations are escalating their use of commercial off-the-shelf (COTS) and other pre-existing components in critical business systems. Attempts to exploit these components through use of traditional engineering approaches that involve defining requirements, formulating an architecture, and then searching for components that meet the specified requirements within the defined architecture have been disappointing.

The Evolutionary Process for Integrating COTS-based systems (EPIC)* redefines acquisition, management, and engineering practices to more effectively leverage the COTS marketplace and other sources of pre-existing components. This is accomplished through concurrent discovery and negotiation of diverse spheres of influence: user needs and business processes, applicable technology and components, the target architecture, and programmatic constraints. EPIC codifies these practices in a structured flow of key activities and artifacts. This alternative approach is a risk-based, disciplined, spiral-engineering approach which leverages the Rational Unified Process® (RUP®).

This document is the first release of a full description of the EPIC framework along with its activities and artifacts. The first release of an overview of EPIC is found in the Software Engineering Institute technical report: CMU/SEI-2002-TR-009. These documents will be updated based on reader's comments and lessons learned from use of EPIC.

* Also known as Information Technology Solutions Evolution Process and Integrating Technology by a Structured Evolutionary Process (ITSEP).

® Rational, RUP, and Rational Unified Process, among others, are trademarks or registered trademarks of Rational Software Corporation in the United States and/or in other countries.



Overview

The Evolutionary Process for Integrating COTS-based systems (EPIC[1]) is designed to support building, fielding and supporting software-intensive systems using available commercial-off-the-shelf (COTS) and other pre-existing components.

Use of commercial off-the-shelf and other pre-existing components[2] is gaining popularity, particularly in communities where the organization's needs match those of one or more commercial information technology (IT) marketplace segments. There is a vibrant market today that delivers COTS software components that range from software development environments to operating systems, database management systems, and increasingly, business and mission applications. COTS components and, to a lesser extent, other pre-existing components, offer the promise of rapid delivery to the end users, shared development costs with other customers, and an opportunity for expanding mission capabilities and performance as improvements are made in the marketplace. Few organizations today can afford the resources and time to replicate market-tested capabilities.

Yet, the promise of using pre-existing components is too often not realized in practice [1, 2, 3]. Many organizations find that COTS-based systems are difficult and costly to build, field, and support. A major cause of this difficulty is that organizations building these systems tend either to assume that components can be simply thrown together or they fall back on the traditional engineering skills and processes with which they are familiar–skills and processes that have been shown not to work in the building of a COTS-based system [4].

[1] Also known as Information Technology Solutions Evolution Process and Integrating Technology by a Structured Evolutionary Process (ITSEP).

[2] Pre-existing components include hardware and software components from the commercial marketplace (i.e., COTS components), the legacy system (a piece of the system being replaced), reuse libraries, or other reuse sources (e.g., freeware, shareware).

Experience [5] shows that the effective use of COTS components demands a new way of doing business: new skills, knowledge, and abilities; changed roles and responsibilities; and different processes–and these changes are not happening. To achieve the benefits of the commercial marketplace while managing the drawbacks, the project must drive an evolving definition of the requirements, the end-user business processes, the architecture, and the cost, schedule, and risk as more is learned about the capabilities of the available COTS components.

Scope

The EPIC approach was developed to help organizations build, field, and support solutions based on COTS and other pre-existing components. Components, as the term is used in EPIC, includes hardware and software components from the commercial marketplace (i.e., COTS components), the legacy system (a piece of the system being replaced), reuse libraries, or other reuse sources (e.g., freeware, shareware).

In EPIC, the fundamental concept is to build, field, and support a *solution* that provides important, useful capability to the organization. Ideally, the scope of a solution is defined so that it can be initially fielded in a period of 6 to 12 months. Many organizations' needs exceed this scope. Where possible, these needs are decomposed into many solutions to be developed in sequence or in parallel–or both. In this case, each solution replaces or augments already fielded solutions with enhanced and added functionality.

A solution integrates

- one or more pre-existing hardware and software components from the commercial marketplace (i.e., COTS components), the legacy system (a piece of the system being replaced), reuse libraries, or other reuse sources (e.g., freeware, shareware)
- any required custom code (including wrappers and "glue")
- appropriate linkage/interface to the broader organization's architecture and external systems
- end user's business processes including any changes necessary to match the processes provided by the integrated components and custom code

EPIC integrates COTS lessons learned, disciplined engineering practice, and end-user business process change management to build, field, and support a solution. EPIC may be used for any solution that is controlled and directed through central management using one or more teams of engineers.

Target Audience

This document is written for the stakeholders responsible for building, fielding, and supporting solutions based on COTS and other pre-existing components. For EPIC, these

stakeholders include the broadest possible group of experts who understand and are empowered to help define

- the mission need [e.g., end users (operators, database administrators, installers), business process owners, customers, sponsors, and assorted subject matter experts]
- systems engineering, integration, and maintenance (e.g., engineers, architects, developers, testers)
- contract and cost analysis (e.g., project office personnel)
- the marketplace and other component sources (e.g., vendors/suppliers, technology experts, industry consultants, legacy maintainers)
- changes to end-user business processes

To be effective, the stakeholders must form a team where each member is committed to sharing individual expertise and is willing to compromise some expectations to accommodate the capabilities of the available components. EPIC provides a framework that links the disparate stakeholders into a coherent team that simultaneously defines and manages tradeoffs among requirements and end-user business processes, system architecture and design, and programmatic factors such as procurement strategy, end-user business process change management, cost, schedule, and risk.

Origins

EPIC evolved from a U.S. Air Force need to meet the challenges of building, fielding, and supporting COTS-based business solutions. To meet this need, EPIC builds on and integrates the work of many others:

- The Software Engineering Institute (SEI) COTS-Based System Initiative has done extensive research on processes needed in the management of COTS-based systems [3,4,6,7,8], engineering techniques for designing and evolving COTS-based systems [9], and evaluation techniques for assessing: COTS-based program risks, suitability of COTS products, and appropriateness of COTS-based system designs[3].
- The Rational Unified Process (RUP) [10, 11] provides a disciplined, risk-based spiral engineering approach that extends the work of Dr. Barry Boehm [12]. The EPIC phases, anchor points, and most artifacts, terms, and descriptions are from the RUP. The RUP also provides essential project management and engineering support activities such as project planning, quality assurance, requirements management, and configuration management.

[3] In addition to the publicly available work, this document builds on *COTS-Based Systems for Program Managers* (tutorial) by L. Brownsword, P. Oberndorf, and C. Sledge; *COTS Product Evaluation* (tutorial) by P. Oberndorf, J. Dean, E. Morris, and S. Comilla-Dorda; *COTS Usage Risk Evaluation* (CURE) by D. Carney, P. Place, and E. Morris; and *Acquisition/Assembly Process for COTS-Based Systems* by D. Carney, P. Oberndorf, P. Place, L. Brownsword, and C. Albert.

Reader's Guide to This Document

This document is designed to explain the basic principles of EPIC so that they can be understood without the need of extensive additional documentation. However, this document makes no attempt to be complete. When the activities come from the RUP, EPIC describes what needs to be done without providing the detailed information necessary for implementation. It is assumed that the reader has direct access to the RUP for detailed explanation and examples of EPIC artifacts. Only when the activities are unique to EPIC are guidelines and artifacts provided. In addition, support activities such as configuration management, requirements management, and quality assurance are captured only in artifact lists.

This document is organized into three sections of chapters, and appendices that provide information relevant to the entire document.

- Section A provides an **Overview** of EPIC. This overview was written so that those who do not need detailed implementation information can learn the fundamental precepts of EPIC. Within this section, Chapter 1 provides insight into the challenges that underlie the definition of EPIC and Chapter 2 summarizes the EPIC framework.
- Section B contains detailed **Phase Descriptions**. Chapters 3 through 6 contain detailed descriptions of each of the four EPIC phases. For each phase, the goal, objectives, exit criteria, and activities are described and the artifacts are listed.
- Section C contains COTS-unique **Guidelines and Artifacts**. Chapters 7 through 13 provide pragmatic considerations to guide a number of COTS component-unique activities such as component evaluation and vendor/supplier relationships.
- Sections D, E, and F include an annotated list of all EPIC **Artifacts**, a **Glossary** of terms related to EPIC, and a list of the EPIC **References** respectively.

This description of EPIC and the overview (Software Engineering Institute technical report: CMU/SEI-2002-TR-009) comprise the first release of EPIC documentation. These documents will be updated based on reader's comments and lessons learned from use of EPIC. Comments or suggestions about the documents are welcome and examples of use of EPIC are solicited. Comments, suggestions, and examples can be sent to the authors at

cbs@sei.cmu.edu

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Chapter 1

Process Drivers

Using pre-existing components, particularly COTS components, introduces significant challenges. These challenges drive the life cycle processes that comprise EPIC.

Building solutions based on incorporating pre-existing components is different from typical custom[4] development in that the components are not designed to meet the project's specification. COTS components are built to satisfy the needs of a market segment. Therefore, an understanding of what the components' functionality how it is likely to change over time must be used to modify the requirements and end-user business processes as appropriate, and to drive the resulting architecture.

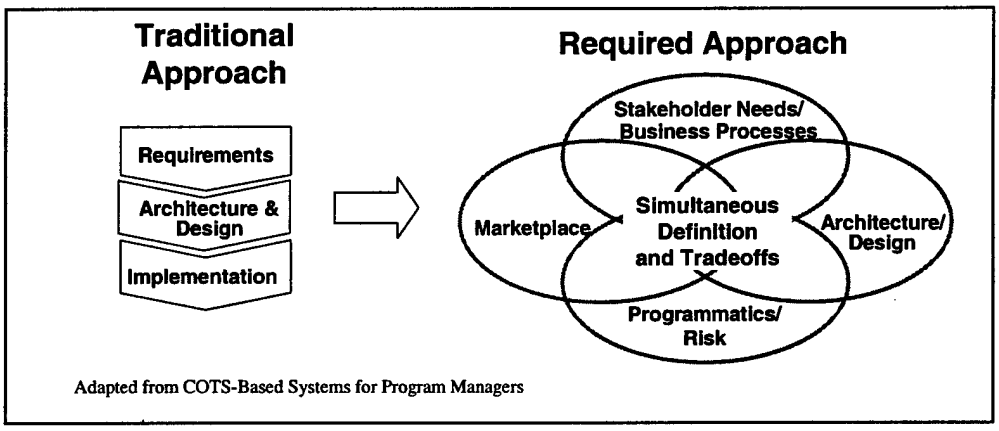


Figure 1. Required Approach for COTS-Based Systems

Key to building solutions based on components is the need to simultaneously define and tradeoff among four *spheres of influence*,[5] as shown in Figure 1. While tradeoffs are common in any engineering endeavor, tradeoffs in this case are driven by the desire to leverage components from the marketplace. This is a fundamental change[6] from alternative process models. Numerous projects have unsuccessfully tried to integrate pre-existing components using the more traditional approach (shown on the left) consisting of defining the

[4] By *custom* development we mean built according to a buyer's specifications.
[5] *Sphere of Influence* is used to represent a set of information with common or related stakeholders, techniques for gathering and managing the information, and the dynamic by which the information changes.
[6] *COTS-Based Systems for Program Managers* (tutorial); L. Brownsword, P. Oberndorf, and C. Sledge.

requirements, then formulating an architecture to meet those requirements, and then trying to fit components into that architecture.

Instead, as shown on the right of Figure 1, an emphasis on balance between four spheres of influence is critical throughout the life of a project. The four spheres represent competing interests that must be considered in forming a viable solution that effectively leverages pre-existing components:

- Stakeholder needs and business processes denotes requirements (including quality attributes such as performance, security, and reliability), end-user business processes, business drivers, and operational environment.
- Marketplace denotes available and emerging COTS technology and products, non-development items (NDI), and relevant standards.
- Architecture and design denotes the essential elements of the system and the relationships between them. The elements include structure, behavior, usage, functionality, performance, resilience, reuse, comprehensibility, economic and technologic constraints and tradeoffs, and aesthetic issues. [11]
- Programmatics and risk denotes the management aspects of the project. These aspects consider the cost, schedule, and risk of building, fielding, and supporting the solution to include the cost, schedule, and risk for changing the necessary business processes.

These four spheres are simultaneously defined and traded through the life of the project because a decision in one sphere will inform and likely constrain the decisions that can be made in another sphere. For example, a stakeholder need may be stated in a way that it cannot be satisfied by any known pre-existing component. Similarly, a potential pre-existing component may not be compatible with the organization's existing infrastructure or use a licensing strategy that would be cost prohibitive. Further, the new release of an already selected component may change the behavior of the system.

Commercial Marketplace-Imposed Constraints

The unique characteristics of COTS components introduce dynamics and specific constraints that must be accommodated. COTS components[7] are

- sold, leased, or licensed to the general public
- offered by a vendor trying to profit from them
- supported and evolved by the vendor, who retains the intellectual property rights
- available in multiple, identical copies
- used without modification of the internals

[7] *COTS-Based Systems for Program Managers* (tutorial). Brownsword, L.; Oberndorf, P.; and Sledge, C.. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999.

Based on these characteristics, the SEI has identified the following attributes [6] of COTS components:

- The marketplace[8], not one system's needs, drives COTS component development and evolution.
- COTS components and the marketplace undergo frequent, almost continuous change.
- Frequency and context of COTS component releases are determined at the discretion of the vendor.
- COTS components are built based on unique architectural assumptions and are not constructed using a universal or consistent architectural paradigm.
- There is at best limited visibility into COTS component internals and behavior.
- COTS component assumptions about end-user processes may not match those of a specific organization.
- "Vendor" is not a new name for subcontractor. Different relationships are required to have insight and to influence component changes.
- COTS components often have unsuspected dependencies on other COTS components.

Derived Requirements

Any process that builds, fields, and supports solutions must

- create an environment where components and the marketplace drive the evolving definition of the solution. EPIC reflects the reality that the ultimate control of critical components has passed from the hands of the project to those of the component suppliers. EPIC focuses on reconciling the diverse expectations of stakeholders with an evolving understanding of the capabilities of available components. Therefore, an environment that facilitates hands-on analysis of the components and continuous negotiation with stakeholders will be required for the life of the project to evaluate new and changed components and their impacts on evolving solutions.
- compose solutions from a diverse range of components. Solutions are built from a combination of components, both hardware and software–and many components are themselves composed from components. Insight into the inner workings of these components will vary (e.g., black, white, and gray box) depending on the source and the intended use of the component. Engineers must infer the behaviors of various component combinations as they integrate the components they buy (and otherwise procure). Hands-on experience is essential with any component critical to the success of the solution in an environment that represents, as closely as possible, the operational use of the component.

[8] *Marketplace* refers to the aggregation of buyers and sellers where goods are offered for sale, lease, or license.

- implement disciplined spiral or iterative systems engineering practices. Spiral development allows the discovery of the critical attributes of the solution through an evolutionary exploration of the highest risk elements of the system and the components available to address them. It also allows frequent and direct feedback from all of the affected stakeholders through evolving prototypes that characterize and mature the architecture while reducing risk in the solution.

- support concurrent and integrated implementation of engineering, business, and procurement activities. The volatility of the COTS marketplace means that decisions about the COTS components used, the structure that accommodates them, and the associated cost, schedule, and risk of the project are made continuously through the life of the solution. In addition, an engineering decision to include a new component is also a decision to acquire the component from its supplier. The engineering processes, management processes, oversight processes, and procurement processes of the project must be coordinated to support the flexibility and negotiation of requirements and iterative definition of the solution.

- balance component obsolescence and solution stability. Due to the volatility of the marketplace, a new component or a new release of a component being used can be introduced at any time during building, fielding, or support of the solution. Engineers must continuously monitor the marketplace through the life of the project to anticipate the changes being made. An environment where new components and releases can be evaluated and their potential impacts assessed is required.

- accommodate a broad spectrum of ways components comprise solutions. Different components can be used in a variety of ways to form a spectrum of possible solutions. Some solutions are composed of a single component or a vertically integrated set of components from one vendor (components designed to work together–e.g., Microsoft Office). The focus in this case is on tailoring the component to work in the target organization. At the other end of the spectrum are solutions composed of multiple components from multiple vendors that must be integrated to provide system functionality. This kind of solution may contain significant components that are custom developed or pieces of legacy systems and may require significant effort to integrate.

- develop and maintain a flexible system architecture as a centrally managed asset. Since the components are "owned" by the suppliers, the framework by which the components are linked to support the organization's needs–the architecture–becomes a key organization asset. With COTS-based systems, continuous, rapid changes driven by new mission needs, component upgrades, and technology advances are facts of life. An architecture that can retain its structure and cohesiveness yet allow the system to respond easily to these changes–an evolvable architecture–becomes an important strategic asset to an organization.

- design *with* components versus design *for* reuse. Building a solution is inherently an act of composition and reconciliation. Designers start with a general set of needs and explore the offerings of the marketplace to see how closely they match the needs, then design an architecture around the pre-existing components. Building systems for reuse,

on the other hand, has come to mean designing and building structures and components that can be used again in related systems. In this case, the designers have direct visibility and control of the components. The nature, timing, and order of the activities and the processes used differ accordingly.

- incorporate activities for changing the end user's business processes. Using components in solutions is not compatible with simply automating a predefined set of business processes. Since components embody the vendor/supplier view of end-user business processes, changes to the end user's business processes must be negotiated based on those inherent in the components under consideration. In EPIC, engineering activities are coordinated with activities for changing the end-user's business process to ready the end-user community for fielding of the solution. The risks and implications of changes to the organizations where the solution is fielded may be significant drivers in the project. EPIC users must coordinate the definition and implementation of end-user business process and organizational changes through the life of the project.
- maintain and document how the component supports the solution. Many components have diverse functionality, not all of which is required in a given solution. It is important to document the functionality applicable to the solution. It is just as important to capture how any excess functionality is handled within the solution–especially any functionality that is blocked or bypassed. Guidance and training on any solution-specific use of the application will have to be considered as part of the deployment artifacts. Furthermore, after fielding, some of these "excess" capabilities will find their way into operational use. The ways in which the components are actually used within the organization must be tracked and captured. This information is important to the evaluation of new component releases after solution fielding, as vendors may make changes in segments of the component that were not originally considered part of the solution.

Chapter 2

Framework

To accommodate the continuous change induced by the COTS marketplace, EPIC applies a risk-based spiral development process. EPIC users manage the gathering of information from the marketplace and the stakeholders and refine that information through analysis and negotiation into a coherent, emerging solution that is embodied in a series of executable representations through the life of the project. Stakeholders actively participate in EPIC as key players in day-to-day negotiations that also continue through the life of the solution. This also ensures their buy-in to the emerging solution.

EPIC is more than a way to select a specific component. Use of EPIC begins with the definition of a need for a new or changed capability and a commitment to provide the resources necessary to identify, acquire, build, field, and support a solution that will deliver that capability. Use of EPIC ends only when the solution is retired or replaced with a new solution. In some instances, the solution will be transitioned to a different organization for support once it has been fielded. The major features of EPIC should also be used by the support organization to protect the investment in components.

Iteratively Converging Decisions

In order to maintain balance between the four spheres, EPIC creates an environment that supports the iterative definition of the four spheres over time while systematically reducing the trade space within each. This allows a decision in one sphere to influence, and be influenced by, decisions in the other spheres.

Initially, as shown at the left of Figure 2, the trade space may be large. There is flexibility for making tradeoffs between the stakeholder needs and end-user business processes, the architecture and design, the offerings of the marketplace and other sources, and programmatics and risk. As EPIC is used to drive toward a refined understanding of the solution, the trade space shrinks. The spheres increasingly overlap as fewer decisions remain in any single sphere that can be made without significant impact on the others.

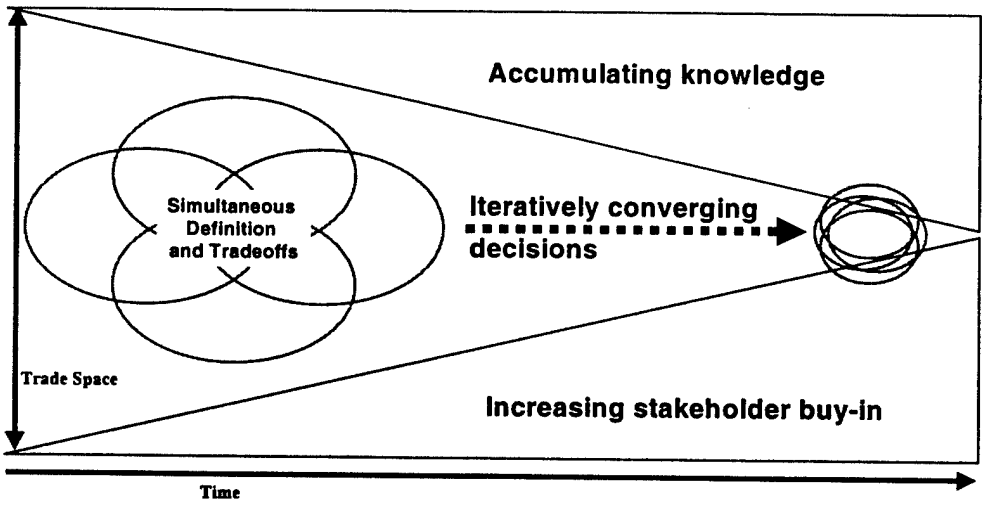


Figure 2. The EPIC Objectives

Accumulating Knowledge

Concurrent with diminishing the trade space, knowledge about the solution must grow at a controlled pace. This knowledge is reflected in the set of artifacts[9] or work products necessary to build, field, and support the solution. Most of the artifacts are started in outline form and are expanded as more information is gathered and refined.

This knowledge includes an increasingly detailed understanding of the following:

- capabilities and limitations of candidate components, the suppliers that produce them, and the marketplace drivers that control them
- negotiated and prioritized stakeholder needs and end-user business processes
- architectural alternatives and integration mechanisms that bind the components together
- implications of the components on the stakeholder needs, the end-user business processes, and the solution architecture
- planning necessary to implement and field the solution (including any needed end-user business changes) and associated cost, schedule, and risk

Keeping knowledge current about the components critical to the solution and the marketplace or other sources that supply components is particularly important. This allows the organization to track trends that may affect the solution over time and keep the volatility of the marketplace in balance with the need for stability for building, fielding, and supporting the solution in operations. Monitoring and evaluation begins at project initiation and continues until the solution is retired.

[9] Artifacts and work products may take a number of forms. Generally, they are not documents.

In some (unfortunate!) cases, marketplace events may invalidate earlier decisions (e.g., support for a component is dropped, a new component is introduced, or a feature is added to the component). While there is no easy resolution for such disruptions, early warning of impending changes will allow alternative decisions to be made in a deliberate and careful way; perhaps even before the trade space narrows.

Increasing Stakeholder Buy-in

While decisions are converging and knowledge is accumulating, the stakeholders must increase commitment to the evolving definition of the solution–and these stakeholders are a broad and possibly disparate group. This will be very difficult for many projects as this commitment is significant, even unprecedented. Active participation from the stakeholders, however, is essential to the success of the solution. Creating an environment that includes the stakeholders (or empowered representatives) directly affected by any change in end-user business processes allows EPIC to quickly resolve discovered mismatches between the available components, the desired end-user business processes, and the stated stakeholder needs while simultaneously demonstrating that the solution can be built within cost and schedule constraints with acceptable risk.

With increased understanding of available components, end-user needs mature and change. The day-to-day involvement of end users is essential to EPIC because the activities that identify, evaluate, and select components will shape the end-user business processes and define the functionality that will be delivered. At the same time, engineering stakeholders ensure that the components considered can be effectively integrated within the broader organization's existing systems to meet required performance parameters. Business analysts must ensure that viable suppliers support the components. Supplier involvement can provide enhanced visibility into the component's capabilities and potential insight for the suppliers into the organization's needs. The continuous negotiation and reconciliation among affected stakeholders leads to a more effective use of components in satisfying the mission.

The stakeholders confirm and increase their buy-in and commitment to the evolving definition of the solution based on an iteratively evolving executable representation of the solution. An executable representation is essential to reduce the risks due to misunderstandings or unforeseen technical and operational factors.

Evolution Through Iterations

EPIC uses a risk-based spiral development process to keep the requirements and architecture fluid as the four spheres of influence are considered and adjusted to optimize the use of available components. Iterations systematically reduce the trade space, grow the knowledge of the solution, and increase stakeholder buy-in. At the same time, each iteration, or spiral, is planned to mitigate specific risks in the project.

Each iteration in EPIC, as shown in Figure 3, begins with the development of a detailed *plan* for the iteration and ends with *assessing* whether or not the objectives in that plan were met. Iteration planning uses the current understanding of risk to establish goals and objectives, and defines the specific tasks as well as the cost, schedule, and resources needed for the iteration.

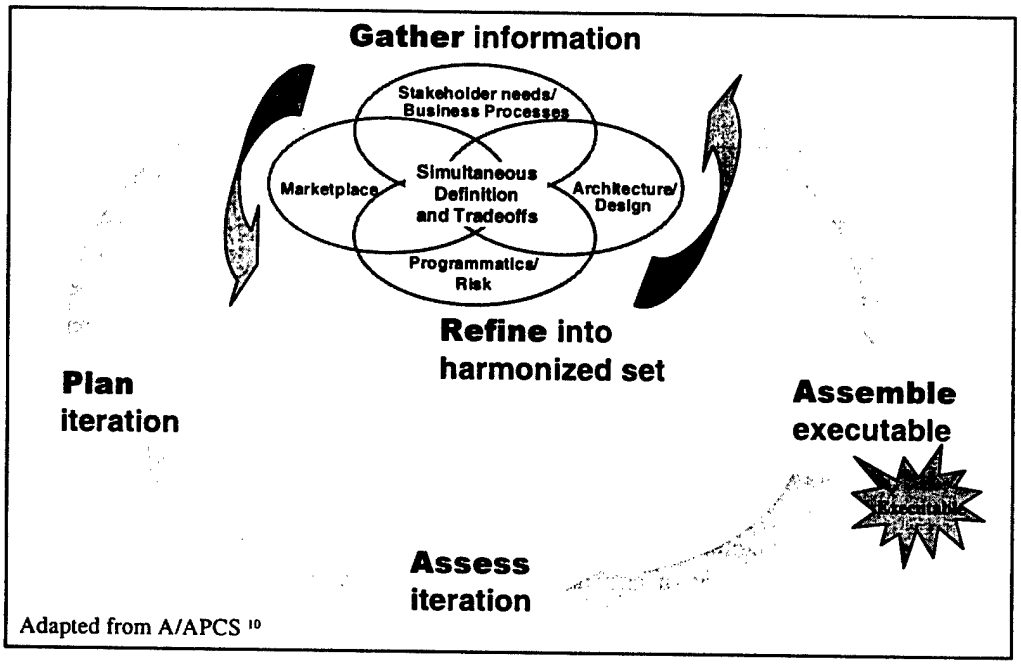


Figure 3. An Iteration in EPIC

Unique to building solutions are a number of inherently chaotic activities.[10] These are the activities that continuously *gather* information from each of the four spheres and *refine* the newly gathered information through analysis and negotiation with affected stakeholders, to form the harmonized knowledge needed to *assemble* an Executable Representation of the solution. This knowledge is captured in artifacts that start at a very high level in early iterations and are expanded through cycles of gathering and refining across iterations as increasingly detailed information is harmonized. It may take many cycles of gathering and refining information within an iteration to produce knowledge sufficiently detailed and harmonized across the four spheres to meet the iteration objectives.

Every iteration assembles an Executable Representation of the solution that exhibits the common understanding of the solution that has been achieved among affected stakeholders to that point and demonstrates the adequacy of the solution to meet the iteration objectives.

While these activities are the same for every iteration, the focus, depth, and breadth of the activities within an iteration are adjusted to meet specific iteration objectives. Each of these activities is further described below.

[10] *Acquisition/Assembly Process for COTS-Based Systems (A/APCS)* by D. Carney, P. Oberndorf, P. Place, L. Brownsword, and C. Albert. To be published.

Plan

Planning is a continuous activity and the planning artifacts are updated with every iteration to reflect an evolving understanding of the solution. Planning occurs at two levels. The first level of planning creates (in the first iteration) and maintains (in each subsequent iteration) a coarse-grained plan for the project. This plan, the Development Plan, describes the capability that will be built and fielded and lays out, in general terms, the number of iterations, key milestones, and resources needed to mitigate risk and deliver the current solution.

The second level of planning creates a fine-grained plan for the current iteration. The Iteration Plan defines the goals and objectives for the iteration and determines the resources (including cost and schedule) necessary to meet them. The objectives for any given iteration are designed to meet high-priority functional needs and to mitigate high-priority risks to the project. (Note: The greatest risk posed to the project will likely be implementing the changes to the end-user business processes to match those in the available components.) Each iteration should be constrained to the amount of work that can be done in a fixed, relatively short (measured in weeks) period of time. Later iterations will build on the results of earlier iterations to produce the desired solution.

Gather and Refine

The gather activities collect the information needed to meet the iteration objectives from each of the four spheres of influence and build representations of the information specific to the type of information gathered. The refine activities synthesize the gathered information through analysis targeted at identifying incomplete information and identifying mismatches among the competing constraints and the potentially divergent classes of expectations. Incomplete information is resolved through gathering additional information. Mismatches are resolved through negotiation between the affected stakeholders. Where possible, end-user business processes are modified and requirements negotiated to allow greater use and leverage of available components.

While it is useful to think about information from these spheres individually because of the different techniques associated with gathering information from each of the spheres, the information gathered from one sphere depends on and drives the information needed from another sphere. In practice, this drives the practioner to gather a little, refine a little, then gather some more, and then refine some more. For example, examination of a component may suggest that end users must select from a number of alternative security strategies. This may require an additional round of stakeholder needs elicitation to determine stakeholder security preferences.

The gather and refine activities produce harmonized artifacts that represent the current agreed-upon state of the solution and include all of the known data and previously accepted compromises necessary to meet the iteration objectives. Mismatches between information from the four spheres are remediated in successive gather and refine cycles using progressively more detailed information until it is determined that all information is sufficiently detailed to meet the objectives for the iteration at hand. It is important not to

allow information in one sphere to be gathered at too low a level of detail relative to the information in the other spheres. This tends to close out opportunities for trades across the spheres.

Assemble

An essential activity in every iteration is the effort to assemble one or more Executable Representations of the current agreed-upon state of the solution. In early iterations, the Executable Representation may be a mock-up of critical stakeholder needs. In later iterations, the Executable Representation is an evolutionary prototype that reflects the architecture and evolves to become the fielded solution. This prototype includes an ability to test the necessary infrastructure and any other systems with which the solution must interact. In addition, the Executable Representation for each phase must be sufficient to prototype the end-user business processes inherent in the solution.

Assess

The assess activities review the iteration to determine whether or not the iteration's objectives were achieved. In addition, a summary of what was learned and what more needs to be learned to mitigate risk is created. These activities also determine whether or not the stakeholders are still committed to proceed to the next iteration.

Phases Focus Iterations

The four spheres continuously evolve through successive iterations. Each iteration is designed to meet specific objectives and will nominally take one to eight weeks to complete. EPIC iterations are managed, as shown in Figure 4, by the four RUP phases (Inception, Elaboration, Construction, and Transition) and associated anchor points[11] (Life-cycle Objectives, Life-cycle Architecture, and Initial Operational Capability).

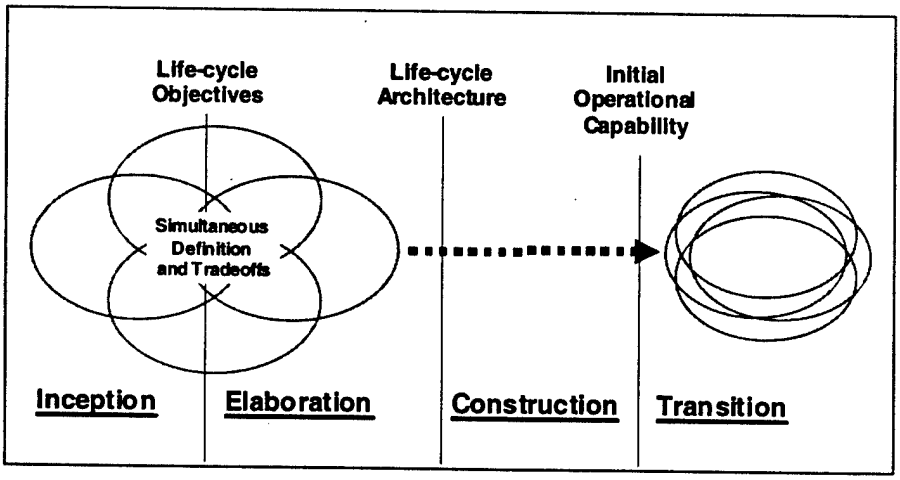


Figure 4. EPIC Phases

[11] RUP calls these lifecycle milestones.

Each EPIC phase consists of one or more EPIC iterations. Iterations in each phase build on and strengthen stakeholder understanding of the available components and each component's impact on requirements and end-user business processes, architecture and design, and the cost, schedule, and risk of implementing the solution.

The EPIC phases accommodate the continuous change induced by the COTS marketplace. Each phase has explicit objectives, activities, artifacts, and phase exit criteria. Each phase ends with an anchor point that provides an opportunity to review progress, ensure continued stakeholder commitment to the evolving solution, and to decide to proceed, change direction, or terminate the project. The four EPIC phases are summarized below and described in detail later in this document.

Inception Phase

The Inception Phase establishes a common understanding among stakeholders of the solution consistent with the organization's longer-term objectives. This common understanding, or scope, defines what the solution will do and why. It provides a basis for planning the technical content of each of the following phases and for estimating the cost and time to develop the solution.

The focus of the Inception Phase is on gathering information from each of the four spheres and capturing that information in the form of project artifacts. Most of the artifacts are started in outline form and will be expanded across later phases as more information is gathered and refined. The disparate stakeholders identify the most critical needs and constraints in each sphere. This information includes

- a high-level understanding of the end-user needs, expectations, and constraints. As solutions are defined, stakeholder requests are challenged to ensure that each need and the implication of not meeting that need are fully understood.
- a market survey to understand the makeup, motivations, and components available in the relevant market segment(s). As components are identified, the suppliers of those components are examined for long-term viability, and limited experiments with the components are conducted to evaluate component suitability.
- the constraints imposed by previous solutions, available technology, and components as well as applicable standards, external interfaces, and any existing systems with which the solution must interact
- the cost and schedule targets for the project, available procurement vehicles for needed components and services, impediments to end-user business process change, and risk

The gathered information is refined through analysis and negotiation with appropriate stakeholders to form one or more feasible, albeit high-level, candidate solutions. A candidate solution is feasible if it describes a useful capability based on available components that can be integrated into the broader organization's architecture, in a reasonable period of time, at affordable cost, and for acceptable risk. The candidate solutions are summarized in an initial business case where identification of the solutions recommended for detailed examination in

the Elaboration Phase is also made. An Executable Representation in each iteration demonstrates the feasibility of the candidate solutions to meet the critical use cases (the primary scenarios of operation).

Because each candidate solution could contain very different combinations of components, and the components address different aspects of the critical use cases, each candidate solution may represent differing stakeholder needs, architecture, user processes, and programmatic decisions. A manageable number (two to four) of feasible candidates will be carried forward into the Elaboration Phase for further definition of the four spheres.

The Inception Phase ends with the Life-cycle Objectives (LCO) anchor point. Whereas in the RUP, LCO means that the requirements are settled sufficiently to form an architecture; in EPIC, LCO means one or more candidate solutions are identified that meet the high-level objectives for the solution and have key stakeholder concurrence. The LCO anchor point reviews the phase exit criteria, determines that the phase objectives have been met, validates stakeholder concurrence on the scope of the solution, and seeks approval to examine the most viable candidate solutions in greater depth.

Elaboration Phase

LCO marks a change in intensity. The basic activities for the Elaboration Phase are the same as those in the Inception Phase, but the level of detail is deeper and the level of resource commitment is significantly higher. The Elaboration Phase achieves sufficient understanding and stability of the requirements, end-user business process, and architecture; selects and acquires components; and mitigates risks such that a single solution is identified that meets a valid organizational need with acceptable cost and schedule.

The focus of the Elaboration Phase is on in-depth hands-on experiments with the candidate solutions by end users and engineers. These experiments and prototypes are conducted in an experimentation facility that represents, as closely as practical, the operational environment. This phase includes further definition of stakeholder needs and end-user business processes based on detailed evaluation of the components. This phase also includes definition and prototyping of the strategy and mechanisms for component integration, data migration, and component tailoring.[12] The focus continues to be keeping the four spheres in balance as greater knowledge of each of the candidate solutions is gained.

When the candidate solutions are sufficiently understood, one solution is selected that will become the basis for the Construction Phase. The selected solution is further amplified, using the experimentation facility, until it is shown that the selected solution has achieved sufficient stability in requirements and architecture as demonstrated in an Executable Representation.

The Elaboration Phase ends with the Life-cycle Architecture (LCA) anchor point when all stakeholders agree that the solution provides sufficient operational value to stakeholders and

[12] *Tailored* means non-source code adjustment necessary to integrate the COTS products into an operational system (e.g., scripts).

can be assembled by the engineers for acceptable cost, schedule, and risk. At this point, all components have been selected and procured, any integration mechanisms to incorporate the components and any other components are validated, and the risk, cost and schedule for completion of the project have been predicted within an acceptable range.

Construction Phase

The focus of the Construction Phase is on preparation of a production-quality release of the selected solution approved at the LCA anchor point that is suitable for fielding. Any custom components needed are developed. Production rigor is applied to component tailoring, integration code or data (including wrappers, glue, data sets, etc.) needed to incorporate pre-existing and custom components, and system testing. Additionally, the Construction Phase includes preparation of necessary support materials, such as installation instructions, version descriptions, user and operator manuals, and other user and installation site support required.

The Construction Phase continues the preparation of the end-user business environment of the target organizations to facilitate the initial fielding of the solution. This preparation includes development of required policies and procedures, restructuring of the organization as necessary, implementation of the changes to the end user's business process for the initial rollout groups, and the establishment of incentives, user groups, and other mechanisms to encourage adoption of the solution.

While every effort is made during the Elaboration Phase to stabilize the solution and to address risks, inevitably some unanticipated changes may occur in requirements, components, and the architecture and design during the Construction Phase. In particular, because of the volatile nature of the marketplace, new versions of the selected components will require detailed investigation as suppliers add, change or remove functionality. Continued monitoring of the marketplace and evaluation of new and changed components is required to anticipate changes and determine an appropriate component upgrade approach.

The Construction Phase ends with the Initial Operational Capability (IOC) anchor point. The IOC anchor point allows stakeholders to verify that a production-quality release of the solution is ready for fielding to at least a subset of the operational users as an initial fielding or beta test.

Transition Phase

The Transition Phase is focused on moving the solution to the user community. This requires that the users attain proficiency in the solution and the end-user business processes that the solution supports, are motivated to use the solution, and are self-supporting in their use of the solution.

The Transition Phase begins with an initial fielding, or beta test of the solution developed in the Construction Phase. Following a decision to make the solution release generally available, the solution will be fielded across the user base. As required, bugs are fixed,

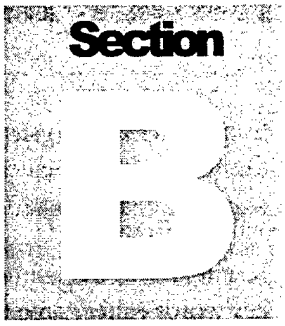
features are adjusted, and missing elements are added to the fielded solution in maintenance releases. Continued monitoring of the marketplace and other sources is required to anticipate changes. Maintaining an experimentation facility for component evaluation to assess the potential impact of any new or changed components is essential.

The Transition Phase encompasses continued support for the solution. The Transition Phase ends when the solution is retired and replaced by a new solution. The activities of this phase are required for support of the solution even if support is provided by an organization different from the organization responsible for its implementation. In this case, it is incumbent on the implementation organization to transfer the knowledge that has been gained in the previous phases and iterations to the support organization.

Summary

EPIC manages the continuous change induced by the use of components through a risk-based spiral development process that actively manages the accumulation of knowledge that is embodied in the solution. EPIC increases stakeholder buy-in to the emerging solution while iteratively converging decisions across the four spheres of influence.

The four EPIC phases are repeated for each solution. Each phase consists of one or more EPIC iterations, each of which produces an Executable Representation. Across the life of a large or complex project, many solutions–often overlapping–are created and retired in response to new technology, new components, and new operational needs.



Phase Descriptions

The following four chapters provide an expanded description of each of the four EPIC phases: Inception, Elaboration, Construction, and Transition. For each phase, this **Phase Description** precedes the **Phase Objectives**, a **Phase Task Overview**, and a checklist for the **Phase Exit Criteria**. The detailed tasks and steps necessary to implement the **Phase Planning Activities**, **Iteration Activities**: *Plan*, *Gather*, *Refine*, *Assemble*, and *Assess* (described in Chapter 2 of Section A), and **Supporting Activities** follow. Each chapter concludes with a summary of the major **Phase Artifacts**.

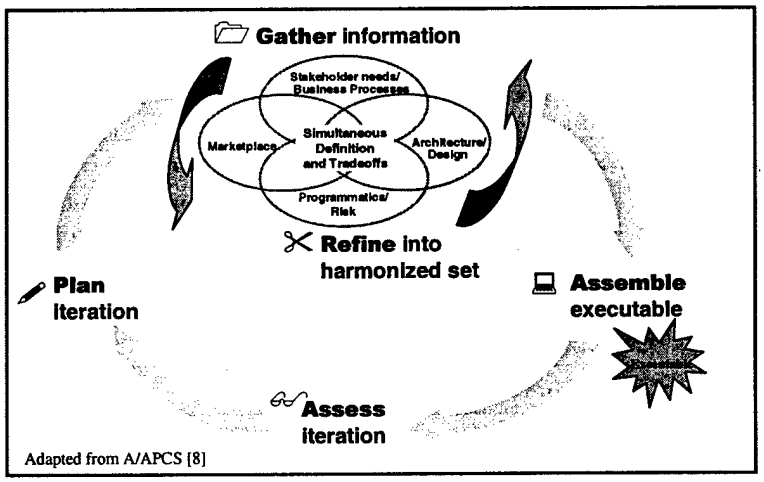


Figure 5. An EPIC Iteration

For ease in using this document, the icon used in Figure 5 is used to indicate the expanded Iteration Activity. This expansion describes the major tasks (indicated with bolded verb phrases) that must be accomplished in each iteration of the phase. Each major task is

further divided into steps indicated by a " ❑ ". As noted in the tasks themselves, many of these steps are executed simultaneously or concurrently. The fact that they are listed in a particular order is *not* to imply that one activity is complete before the next is begun.

Key words to the left of each step are provided for ease in navigating the document. These key words either name the artifact used by the subsequent steps, or define a major topic area that describes steps that will lead to one or more artifacts. Artifacts, which can be thought of as interim work products of the process, may take a number of forms–generally, they are not documents. Artifact names are indicated in the text using a special font. Tips provide an idea of when or to what depth a step should be implemented and, when used, are prefaced with a "❖".

Phase Activities and Tasks

While every iteration has the same activities, the emphasis within each activity, and the tasks that support that activity, may change depending on the phase. Figure 6 summarizes the tasks in each of the major iteration activities across the four EPIC phases.

Inception	Elaboration	Construction	Transition
Plan the Iteration			
Build a detailed plan for the iteration			
Update the Development Plan for the project			
Gather...			
... an Understanding of Stakeholder Needs and End-User Business Processes			
Update or create a business model	Update and expand the business model	Update and expand the business model as necessary	
Capture the critical behaviors of the solution	Capture the significant behaviors of the solution	Capture the behaviors of the solution	Update the behaviors of the solution as needed
... an Understanding of Architecture and Design			
Determine architectural context	Amplify the architectural context	Review, and update as needed, the architectural context	Monitor the architectural context
Identify architectural alternatives	Amplify the architectural alternatives contained in solution(s)		
... an Understanding of Marketplace and Other Sources			
Identify relevant component sources	Monitor relevant component sources	Monitor relevant market segments	
Characterize available components	Evaluate applicable components	Characterize component changes	
... an Understanding of the Programmatics and Risks			
Identify management information	Update management information		

Figure 6. Iteration Tasks by Phase

Inception	Elaboration	Construction	Transition
Identify procurement needs and opportunities	Update procurement needs and opportunities		
Identify implications of potential changes to the end user's business process	Amplify implications of potential changes to the end user's business process	Monitor implications of changes to the end user's business process	
Identify risks	Update risks		
Refine the Understanding of the Solution			
Synthesize information in candidate solutions	Identify and resolve mismatches from the synthesis of new information		
Analyze and negotiate mismatches for each candidate solution			
Characterize each candidate solution	Amplify the solution(s)	Update the solution if needed	
Assemble an Executable Representation			
Build and test proof of concept(s)	Build and test an architectural prototype	Build and test the solution	Build and test releases of the solution
Prototype the needed changes to the end user's business process		Implement the needed changes to the end user's business process	
		Make any needed existing infrastructure and external interfaces changes	
Assess the Iteration			
Assess the Executable Representation	Assess the architectural prototype for the solution(s)	Assess the solution	
Update the information about the solution			
Determine lessons learned from iteration			
Assess the phase, if the iteration completes the phase			Review all phases if the iteration completes the soluiton

Figure 6 (Continued). Iteration Tasks by Phase

A summary of **Supporting Activities** follows the Iteration Activities in each phase description. Supporting Activities include tasks that must be accomplished during iterations within the phase, but may not be part of every iteration. Some of these activities are summarized in Figure 7. The activities associated with project monitoring and control, and many of the supporting process activities described in the Development Plan (such as requirements management or configuration management) are not explicitly addressed here, but are vital to the success of the project.

Inception	Elaboration	Construction	Transition
Monitor project status			
Prepare experimentation facility		Maintain the experimentation facility	
Update and create contracting vehicles as necessary			

Figure 7. Supporting Tasks by Phase

Major Artifacts

Each Phase Description concludes with a listing of the major artifacts used. Each of the artifacts is described in Appendix A, and is summarized in Figure 8 with its primary use in EPIC.

TO CHARACTERIZE END-USER BUSINESS PROCESSES AND STAKEHOLDER NEEDS

- Current Business Use-case Model
- Current Business Object Model
- Target Business Use-case Model
- Target Business Object Model
- Glossary
- Stakeholder Requests
- Solution Requirements Specification
- Use-case Model
- Use Cases
- Supplementary Specification

TO CHARACTERIZE THE MARKETPLACE AND OTHER SOURCES

- Market Segment Information
- Component Dossier (for each examined component)
- Component Screening Criteria and Rationale

TO CHARACTERIZE THE ARCHITECTURE AND DESIGN

- Solution Vision
- Architecture Document
- Design Model
- Executable Representation (s)
  - Implementation Model
- Test Set Artifacts (includes the Test Plan)
- Deployment Artifacts
  - End-user Support Materials (optional in first two phases)
  - Release Notes (required in Transition)
  - Training Materials (required in Transition)
  - Installation Artifacts (required in Transition)

DEVELOPMENT PLAN ELEMENTS TO CHARACTERIZE PROGRAMMATICS AND RISK

Management Process

- Project Plan
- Iteration Plans

Project Monitoring and Control

- Requirements Management Plan
- Schedule Control Plan
- Budget Control Plan
- Quality Control Plan
- Reporting Plan
- Measurement Plan

Risk Management

- Risk Management Plan

Technical Process Plans

- Development Case
- Infrastructure Plan
- Solution Acceptance Plan

Supporting Process Plans

- Configuration Management Plan
- Evaluation Plan
- Documentation Plan
- Quality Assurance Plan
- Problem Resolution Plan
- Vendor/Supplier Relationship Plan
- Process Improvement Plan

ADDITIONAL ARTIFACTS THAT CHARACTERIZE PROGRAMMATICS AND RISK

- Business Case (includes business context, success criteria, financial forecast)
- Business Process Change Management Plan
- Risk List
- Deployment Plan
- License Agreements
- Iteration Assessments (one/iteration)
- Review Record (one/phase)

Figure 8. Master Artifact List

Differences from the RUP

For people who are experienced with the RUP, this section lists some of the major ways that the RUP has been modified for EPIC. This list is not intended to be exhaustive, but to provide highlights of the changes made.

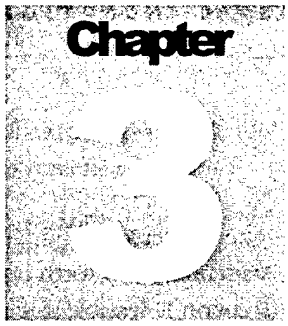
To meet the demands of solutions, EPIC refined the RUP workflows as follows:

- Requirements must stay fluid until the component implications are understood–often until well within the Elaboration Phase.
- The analysis and design activities must start sooner–in parallel with the requirements activities.
- The project management activities include the management of vendor/supplier relationships.
- Business modeling is mandatory rather than optional.
- Activities are added to monitor the marketplace and evaluate candidate components.
- Business, contracting, and organizational change activities are integrated throughout.

In addition, EPIC expands the RUP in the following areas:

- LCO has been redefined to allow multiple candidate solutions to proceed to the Elaboration Phase.
- Iterations are more chaotic to allow for simultaneous gathering and refining of information in all four spheres.
- An experimentation facility is essential across all phases to evaluate new and changed components in the context of the evolving solution.

Readers who refer to RUP sources should note that the RUP use of *system* or *product* is generally equivalent to "solution" in EPIC.



Inception Phase

The goal of the Inception Phase is to achieve concurrence among affected stakeholders on the life-cycle objectives for the solution. The Inception Phase establishes feasibility of the solution through the business case that shows that one or more candidate solutions exist.

Phase Description

The Inception Phase establishes a common understanding among stakeholders of the solution consistent with the organization's longer-term objectives. This common understanding, or scope, defines what the solution will do and why. It provides a basis for planning the technical content of each of the following phases and for estimating the cost and time to develop the solution.

The focus of the Inception Phase is on gathering information from each of the four spheres and capturing that information in the form of project artifacts. Most of the artifacts are started in outline form and will be expanded across later phases as more information is gathered and refined. The disparate stakeholders identify the most critical needs and constraints in each sphere. This information includes

- a high-level understanding of the end-user needs, expectations, and constraints. As solutions are defined, stakeholder requests are challenged to ensure that each need and the implication of not meeting that need are fully understood.
- a market survey to understand the makeup, motivations, and components available in the relevant market segment(s). As components are identified, the suppliers of those components are examined for long-term viability, and limited experiments with the components are conducted to evaluate component suitability.
- the constraints imposed by previous solutions, available technology, and components as well as applicable standards, external interfaces, and any existing systems with which the solution must interact

- the cost and schedule targets for the project, available procurement vehicles for needed components and services, impediments to end-user business process change, and risk

The gathered information is refined through analysis and negotiation with appropriate stakeholders to form one or more feasible, albeit high-level, candidate solutions. A candidate solution is feasible if it describes a useful capability based on available components that can be integrated into the broader organization's architecture, in a reasonable period of time, at affordable cost, and for acceptable risk. The candidate solutions are summarized in an initial business case where identification of the solutions recommended for detailed examination in the Elaboration Phase is also made. An Executable Representation in each iteration demonstrates the feasibility of the candidate solutions to meet the critical Use Cases.

The Inception Phase begins with defining a Development Plan (if it was not created previously) that captures (frequently by reference) the project structure and all of the engineering and management processes that will be used by the project. If necessary, the Development Plan will be updated to meet the more specific demands of the current phase of the project. All elements of the Development Plan are updated iteratively throughout the project. The Development Plan includes the Project Plan and the Iteration Plans. The Project Plan first describes the overview of how the project will be implemented. It describes the starting point for the project; establishes goals, key events, and milestones; schedules the general activities across the four phases; and identifies the necessary resources (including staffing). These are estimates and they are expected to change through execution of the project.

Expanding the Project Plan for this phase is the next activity of the Inception Phase. This expanded plan will lay out, in general terms, the risks to be addressed in this phase, the number of iterations thought necessary to mitigate those risks, and the resources allocated to this phase. Similarly, before the end of this phase, the Project Plan will be expanded for the Elaboration Phase. The Project Plan, expanded for the current phase, becomes the starting point for the detailed Iteration Plan that is created at the start of each iteration.

In this phase, one or more high-level or abstract candidate solutions are formed from the disparate information gathered from each of the four spheres of influence. As solutions are defined, stakeholder requests are challenged to ensure that each need and the implication of not meeting that need are fully understood. In addition, suppliers and components are examined to prevent consideration of those that do not meet functional needs, cannot support integration into the broader organization's architecture (including interfaces to any existing systems), or do not meet project cost and schedule constraints. Examining suppliers and components may require bringing critical components into a local experimental facility to try them out.

A Solution Vision is formed for each candidate solution that captures major features, quality attributes, constraints (including external interfaces), and concept of operations. In addition, potential change scenarios (e.g., forecasts of components and technology, mission, externally driven architecture changes) are captured. Each candidate solution is assessed to determine its expected benefits, risk, cost, and schedule. These assessments are captured in the Business

Case, which describes each candidate solution and identifies the solutions recommended for detailed examination in the Elaboration Phase. If the Inception Phase reveals an overly large number of candidate solutions, then screening is conducted to reduce the number of solutions for consideration. This screening is also described in the Business Case.

Because each candidate solution could contain very different combinations of components, and the components address different aspects of the critical use cases, each candidate solution may represent differing stakeholder needs, architecture, user processes, and programmatic decisions. A manageable number (two to four) of feasible candidates will be carried forward into the Elaboration Phase for further definition of the four spheres.

There is at least one iteration in the Inception Phase. The actual number of iterations required would depend on the complexity of the candidate solutions and on the magnitude of mismatch between the initial expectations of the stakeholders and the capability that can be fielded with reasonable cost, schedule, and risk. Each iteration will be composed of essentially the same activities (as described previously and expanded later in this chapter). The first iteration will be at a relatively high level of detail, and further iterations expand the level at which each activity is conducted and/or mitigate other high-priority risks.

An Executable Representation is assembled in each iteration to show the agreements made in meeting the iteration's objectives. At least one executable representation in this phase (very likely a mock-up) will demonstrate the feasibility of the candidate solutions to meet the critical use cases (the primary scenarios of operation).

Towards the end of the Inception Phase, preparations begin for implementing the Elaboration Phase. These preparations include making sure that the components contained in the candidate solutions are available for exploration. Here, "available" means that license agreements, contracts, or any procurement vehicles necessary to evaluate the components in the Elaboration Phase are in place. In addition, if one does not already exist, an experimentation facility must be ready for receipt of the components. This facility should replicate, as closely as possible or practical, the operational environment (including interfaces to any important external systems).

The Inception Phase ends with the Life-cycle Objectives (LCO) anchor point. Where in the RUP, LCO means that the requirements are settled sufficiently to form an architecture; in EPIC, LCO means one or more candidate solutions are identified that meet the high-level objectives for the solution and have key stakeholder concurrence. At the LCO anchor point,

- the phase exit criteria are reviewed
- the phase objectives are determined to have been met
- stakeholder concurrence on the scope of the solution is validated
- the expanded plan for the Elaboration Phase (in the Project Plan) is reviewed and resourced
- the recommendation to examine the most viable candidate solutions in greater depth is approved

Phase Objectives

Establish a common understanding of the scope of the solution and its boundary conditions, including interfaces to systems outside the boundary of the solution.

Outline viable candidate architectures.

Differentiate the critical Use Cases of the solution, the primary scenarios of operation that will drive the major tradeoffs.

Identify and evaluate the relevant segments of the commercial marketplace and other sources for components and vendors/suppliers and negotiate tradeoffs among critical Use Cases, the candidate architectures, cost, schedule, and risk.

Exhibit and demonstrate at least one candidate solution against the critical Use Cases.

Estimate cost, schedule and potential risks for each candidate solution.

Determine potential changes to the end users' business process and the tolerance for and inhibitors to implementing those changes across the organization.

Establish a plan for acquiring any components and services needed for this solution.

Phase Task Overview

Phase Planning Activities
- *Plan the Project*

Iteration Activities
- *Plan the Iteration*
  - Build a detailed plan for the iteration
  - Update the Development Plan for the project
- *Gather ...*
- *... an Understanding of Stakeholder Needs and End-User Business Processes*
  - Update or create a business model
  - Capture the critical behaviors of the solution
- *... an Understanding of Architecture and Design*
  - Determine architectural context
  - Identify architectural alternatives
- *... an Understanding of Marketplace and Other Sources*
  - Identify relevant component sources
  - Characterize available components
- *... an Understanding of the Programmatics and Risks*
  - Identify management information
  - Identify procurement needs and opportunities
  - Identify implications of potential changes to the end user's business process
  - Identify risks
- *Refine the Understanding of the Solution*
  - Synthesize information in candidate solutions
  - Analyze and negotiate mismatches for each candidate solution
  - Characterize each candidate solution
- *Assemble an Executable Representation*
  - Build and test proof of concept(s)
  - Prototype the needed changes to the end user's business process
- *Assess the Iteration*
  - Assess the Executable Representation
  - Update the information about the solution
  - Determine lessons learned from iteration
  - Assess the phase, if the iteration completes the phase

Supporting Activities
- Monitor project status
- Prepare experimentation facility
- Update and create contracting vehicles as necessary

Phase Exit Criteria

Status	Exit Criteria
	Affected stakeholders concur that the scope of each of the candidate solutions is a feasible and represents a useful capability.
	Critical mismatches between stakeholder needs and component capabilities are negotiated and represented in critical use cases.
	Cost/schedule estimates, project tasks, risk, and engineering processes for each candidate solution are credible.
	The risks for each candidate solution are understood, fall within an acceptable range, and mitigations are identified for critical risks.
	The potential changes to the end user's business process have been identified for each candidate solution, with stakeholder agreement to implement the changes, should the solution be selected.
	The depth and breadth of an Executable Representation (e.g., mockup, architecture prototype) demonstrate the defined scope of each candidate solution.
	The project has initiated relationships with key vendors and suppliers to provide needed insights into component capabilities and directions.
	Any differences between actual resource expenditures versus planned expenditures for this phase are understood, and corrective actions are included in the Project Plan.
	The Development Plan has been updated. (The plan for the Elaboration Phase is sufficiently detailed and accurate, and is backed up with a credible basis for all estimates.)
	The experimentation facility is sufficient to evaluate the impact of candidate components and the candidate solutions within the broader context of the organization's architecture in the Elaboration Phase.
	The License Agreements, contracts, and procurement vehicles for needed components and services are in place for the Elaboration Phase.

Phase Planning Activities

Plan the Project

Before the Inception Phase starts, or at the beginning of the phase, planning starts with creating a Development Plan for the project. The Development Plan[13] references the entire set of major planning artifacts that describe how the project will be executed. There are many plans referenced, but these plans will vary in formality depending on the size and needs of the project. The planning information may not be very detailed at this point with the exception of the detailed planning for the Inception Phase. Updates or revisions to the Development Plan will be made based on lessons learned in engineering and management activities in subsequent iterations as plans are made for the next iteration. In addition, at the end of each phase, a detailed plan for the Elaboration Phase is prepared and the Development Plan is updated accordingly.

Project organization

- Define roles and responsibilities internal and external to the project.
- Define the project's organizational structure.

Risk List

- Elicit and capture technical and non-technical risks associated with the complexity of the project, the business domain to be explored, and any external constraints (e.g., cost, schedule, policy) in the Risk List.
  - Include representative stakeholders (including architects, engineers, infrastructure managers, representative end users, business process owners, program managers) in describing a set of risks that is as comprehensive as possible.
  - Methods will range in formality from structured brainstorming, to more formal methods such as the SEI Software Risk Evaluation or MITRE's Risk Matrix [13, 14, 15].

Project Plan

- Determine primary objectives and exit criteria for each phase. Define end dates for each phase. Develop a work breakdown structure for the project. Estimate the number of and objectives for iterations in each phase.
  - At project start, it will be difficult to predict with any certainty the real number of iterations. It is important, however, to "rough out" the objectives or focus of iterations to address project risks as they are known.

[13] The "Development Plan" is an RUP artifact. This artifact describes the basic engineering and management processes the project needs to acquire, build, field, and support a solution–its use is not limited to "development" activities.

- Define each solution release, with the nature of the release (e.g., demo, alpha, beta, full, bug-fix).
  - Early in the project, little may be known. As the iterations progress and greater detail is known, the release planning information is updated.
- Identify project staff resources and skills required by phase and iteration.

Monitoring and control

- Develop the Requirements Management Plan. Describe how the stakeholder needs will be captured and managed in requirements documentation. Specify the information and control mechanisms to be collected and used for measuring, reporting, and controlling changes to the project requirements.
  - Include requirement types and their respective requirement attributes.
- Define an approach for monitoring progress against planned schedule, including actions for correcting discrepancies, and capture in the Schedule Control Plan.
- Define an approach for monitoring spending against the budget, including actions for correcting discrepancies, and capture in the Budget Control Plan.
- Define quality control methods for project deliverables and when they will be applied, and capture in the Quality Control Plan.
- Define the reports that will be generated for the project, including their frequency and distribution, and capture in the Reporting Plan.
- Define project measures and the project's Measurement Plan.

Risk management

- Describe how the project will manage its risks. Detail the risk management tasks that will be carried out, assign responsibilities, and identify any additional resources required for the risk management activity. Capture in the Risk Management Plan.

Technical process planning

- Create a high-level Development Case for the project. The Development Case focuses on what to do and how to do it. It provides an overview of the processes to be followed, so everyone on the project can understand them.
  - As the process unfolds, lessons are learned during the process itself, which are used by the process engineer as feedback to improve the process.
- Define the project's technical standards and guidelines (e.g., methods, notation, tools, and techniques) in support of the Development Case.
  - Included here are Business Modeling Guidelines, User Interface Guidelines, Use-case Modeling Guidelines, Design Guidelines, Programming Guidelines, Test Guidelines, and a Manual Style Guide.
  - Guidelines should clearly differentiate the role of component evaluation and test for the project. While component testing comparable to unit testing must be accomplished, component evaluation is more than determining whether or not a component meets a fixed set of criteria. Component evaluation must also include the gathering of information that will frame the tradeoffs between the marketplace and the other spheres of influence.
- Develop an Infrastructure Plan for the engineering and experimentation facility environments.
  - Develop a plan for an experimentation facility for the project with high fidelity for the current phase (it will be updated with each iteration).
- Develop a Solution Acceptance Plan that documents the minimum criteria for stakeholder acceptance of the delivered solution and how these criteria will be evaluated.

Supporting processes

- Develop the project's Configuration Management Plan.
- Develop an Evaluation Plan for the project that covers the techniques, criteria, metrics, and procedures used for evaluation of the solution–this includes walkthroughs, inspections, and reviews.

- Develop a Document Plan that describes what documents will be produced for the project, when they are needed, and who will review them.
  - ❖ This does not apply to every artifact, but usually only to those that are needed external to the project.
- Develop the project's Quality Assurance Plan.
- Develop the project's Problem Resolution Plan.
- Define a strategy for procuring the components and services needed for this project. Capture in the Vendor/Supplier Relationship Plan.
- Develop the objectives and approach for monitoring and influencing component directions of critical vendors and suppliers. Capture in the Vendor/Supplier Relationship Plan.
- Develop the project's Process Improvement Plan.

Test planning

- Develop the plans and strategies for test. Capture them in the Test Set Artifacts.
  - ❖ Determine the level of confidence required in the solution and the level of testing needed to achieve it.

Plan for fielding

- Develop a Deployment Plan for initial and full fielding.

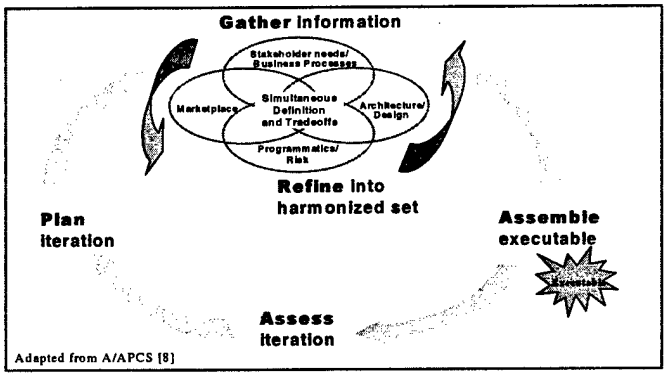
End-user business process change planning

- Create an initial Business Process Change Management Plan.
  - ❖ Initially, the plan will need to accommodate each candidate solution.

Iteration Activities

ITERATION ACTIVITIES KEY

- Plan (Iteration)
- Gather
- Refine
- Assemble
- Assess



Plan the Iteration

Plan the iteration in detail at the beginning of each iteration. The known risks captured in the Risk List are a key consideration for defining the iteration objectives and activities. The information (and the level of detail of that information) to be **gathered** about each of the spheres of influence is determined by the iteration objectives. The iteration objectives also determine what analysis must be done to **refine** the information from each of the spheres. Any lessons learned and new risks discovered from previous iterations are incorporated.

Build a detailed plan for the iteration

The detailed plan for the iteration is based on the current Risk List and the most critical functionality. Risk is a key discriminator in deriving objectives for the iteration. The highest priority risks are mitigated as early as possible. However, addressing risk is balanced with ensuring that the critical functions and services a solution must provide are addressed early.

Iteration Plan

- Refine the scope of the iteration and the goals and objectives that were planned in the Project Plan for this iteration to reflect any changes since the plan was last updated.
- Define objectives for the success of the iteration.
  - ❖ These objectives will provide focus for all activities in the iteration and will be used at the end of the iteration to decide whether the iteration was a success.

- Define the requirements for the Executable Representation needed to demonstrate the iteration objectives.
  - An Executable Representation prototypes the architecture and shows feasibility and consistency with the broader organization's architecture.
  - This prototype is in addition to one or more *exploratory*, throwaway prototypes to mitigate specific risks such as design and requirements tradeoffs, component feasibility analyses, or demonstrations of key scenarios to certain stakeholders conducted as part of **gather** or **refine**.
  - Depending on the nature of the differences between candidate solutions and the specific objectives of the iteration, there may have to be more than one prototype in an iteration.
- Identify the tasks required to achieve the iteration objectives and the specific artifacts that must be developed or updated.
- Complete a detailed work breakdown structure to show how the work that must be done within the iteration is allocated and what resources are necessary to do the work.
- Determine milestones (events and dates) that are important to the iteration.

Update the Development Plan for the project

Development Plan

- Revise and update the Development Plan.
  - The plan should be updated to reflect changes to the technical or programmatic baseline, to reflect changes in personnel availability or skills, to reflect the changes necessary to accommodate a particular set of components, or simply to reflect a new approach to meeting the identified needs.

Gather Information

There is a **gather** activity, comprising discrete tasks, that collects the information needed to meet the iteration objectives from each of the four spheres of influence, and builds representations of the information specific to the type of information gathered.

The information gathered from one sphere depends on and drives the information needed from another sphere. Yet, it is useful to think about information from these spheres individually because the nature of the information in each sphere is fundamentally different, the dynamics of the information are different, and the techniques associated with gathering information from each of the spheres are different. In practice, this drives the process to gather a little, refine a little, then gather some more, then refine some more. The **gather** tasks within each of the four spheres, therefore, occur concurrently. In addition, the **refine** tasks for this iteration manage the interaction among, and integration of the information from, the **gather** tasks. The **gather** and **refine** activities will continue to cycle until the information becomes sufficiently detailed to meet iteration objectives.

Gather an Understanding of Stakeholder Needs and End-User Business Processes

The emphasis in this phase is to identify, at a high level, the business and stakeholder needs that will determine the scope of the solution and to understand the structure and dynamics of the organization in which the solution will be fielded. Information is drawn from the stakeholders who will use or depend on the new solution. These stakeholders include beneficiaries of a legacy solution being replaced; operators, managers, or other users of the new solution; stakeholders of systems that interface with the new solution; and the project manager and customer of the new solution.

Update or create a business model

The goals of the business modeling are to understand the processes the end-user organization uses to perform its mission, and to provide stakeholders with a common understanding of the behavior of the *business* in which the solution will operate. Models must be created for both the *target* business and the *current* business. The two models are developed in parallel as one provides information and insights needed for the other. The target end-user business processes will be revisited and potentially modified during **refine** as information from the other **gather** areas is analyzed.

The success of the solution will ultimately be determined by measurable improvements in business behaviors. Goals for business improvement should be linked to the business model.

Mission

- Characterize the mission and strategic direction of the end-user organization.
- Describe potential changes and forces for change envisioned for the end-user organization and its mission over time.
- Identify the business goals of the end-user organization that relate to, or are affected by, the solution.

Stakeholders

- Identify stakeholders (including the various types of users) who have a vested interest in the outcome of the solution.

Current fielding capabilities

- Characterize the current operational environment for each of the relevant stakeholders who will use some aspect of the solution as well as their current capabilities and available resources for system upgrades or major enhancement.
  - ❖ The end user is able to easily incorporate certain kinds of changes while other changes are more difficult. This may affect the definition of the solution and will drive the planning for fielding. The nature of which changes are easy or hard will depend on the culture of the organization.

Change drivers

- Understand why a change in the current environment is thought to be necessary. To help characterize the issues that have to be considered, determine the root causes of the problems or shortcomings in the current business environment.

Business measures

- Define measurable improvement goals [16] for the business.
  - ❖ If necessary, breakdown the business goals into manageable sub-goals.
- Create the metrics that will be used to evaluate business improvements from the solution implementation.
  - ❖ Identify measures that will characterize, in quantitative terms, whether or not the solution is successful in achieving the desired business objectives. These may be stated in terms of measuring productivity gains, cost avoidance or savings, throughput, etc.

Business modeling objectives

- Determine the objectives for modeling the *current* and *target* end-user business processes and the level of detail to which the modeling effort should go to meet those objectives.
  - ❖ Model the *current* end-user business processes in sufficient detail to determine the implications of implementing end-user business processes implied in candidate solutions.
  - ❖ Model the *target* end-user business processes in sufficient detail to differentiate the important capabilities required in candidate components.
  - ❖ Avoid modeling for modeling's sake!

Current Business Use-Case Model

- Describe current high-level business functions, roles, deliverables, and links to other businesses or systems by identifying the business actors and the business use cases. Capture actors and use cases in a Current Business Use-case Model.
  - ❖ The actors are the users and systems that are external to the business and interact with the business, e.g., a customer of a auto dealership. Business use cases are sequences of events by which actors interact with the business elements to get their job done, e.g., "buy a car."

Current Business Object Model

- Develop a Current Business Object Model that shows the business entities and how these entities provide the functions necessary to realize the current business use cases.
  - ❖ Examples of business entities are payroll clerk or paycheck.

Target Business Use-Case Model

- Describe target business functions, roles, deliverables, and links to other businesses or systems by identifying the actors in the business and the business use cases they will use. Capture actors and use cases in a Target Business Use-Case Model.

Target Business Object Model

- Develop a Target Business Object Model that shows the entities and how they will provide the functionality necessary to realize the target business use cases.

Glossary

- Develop the initial Glossary as part of building the business model by capturing the important terms relevant to the business.
  - Include terms and definitions that are unique to this application context, acronyms, abbreviations, and organization-specific shorthand that are necessary to understand the application area.

Capture the critical behaviors of the solution

The purpose of this task is to collect and elicit information from the stakeholders to understand what their needs really are.

The collected Stakeholder Requests can be regarded as a "wish list" that will be used as primary input to defining the high-level features of the solution, as described in the Solution Vision (which is defined by the **refine** activity), the Use Cases, and the Supplementary Specification. Eliciting Stakeholder Requests and needs in parallel with development of the Target Business Use-case Model and Target Business Object Model is often effective.

Use Cases and the Use-case Model are used to capture the functional behavior of the solution. In the Inception Phase, Use Cases will only be developed for those considered critical (i.e., those Use Cases that will drive key component choices and architectural decisions). The objective of the Use-case Model is to understand the required behavior of the solution (in contrast to Business Use-case Model and Business Object Model that focus on the behavior of the business).

The Supplementary Specifications capture the non-functional behaviors and quality attributes of the solution that are not readily captured in the Use Cases (for example, legal and regulatory requirements and application standards; quality attributes of the solution, including usability, reliability, performance and supportability; and other requirements such as operating systems and environments, compatibility requirements, and design constraints.)

The characterizations of needs for the solution will be revisited during **refine** as information from the other **gather** areas is analyzed.

Stakeholder Requests

- Collect Stakeholder Requests (the raw input and "wish lists" from the various stakeholder groups) for candidate solutions through appropriate elicitation techniques.
  - Stakeholder Requests are the input for determining actual needs.

- ❖ Requests are often phrased as "we need ..." by stakeholders. Requests may come in the form of formal documents such as mission need statements or "requirements specifications." Regardless of the form, all should be treated as requests until analyzed as part of determining stakeholder needs (see next activity).

- ❖ The elicitation techniques used vary based on application, engineering team skill, stakeholder skill, scale of problem, criticality of the solution, and uniqueness of the application.

- ❖ Stakeholder needs specify "what" a solution must do, *not* "how" the solution will meet the need.

- ❑ Challenge Stakeholder Requests to form "must have" needs (those necessary to meet the end-user mission). These "must have" needs should be separated from "important to have" needs or "nice to have" needs (those that enhance but are not mandatory to meet the mission).

  - ❖ "Must have" needs will form the basis for identifying the most critical use cases.

  - ❖ Use the Target Business Use Cases to challenge and clarify requests to form statements of need.

Use-case model

- ❑ Identify all Use Cases (that can be identified at the current level of understanding) for the solution(s).

- ❑ Prioritize Use Cases.

Critical Use Cases

- ❑ Find actors and Uses Cases from the Target Business Use-case Model and Target Business Object Model.

  - ❖ Identify actors and actions that are critical to the solution(s).

- ❑ Provide detail for the critical Use Cases to the level needed to meet the iteration objectives.

  - ❖ The *critical* Use Cases are those that drive the solution's functionality and shape the major design tradeoffs.

Use-case mechanisms

- Identify the mechanisms and services that are needed by the critical Use Cases.
  - Examples include data persistence, security features, distribution and concurrency, transaction management, and fault tolerance.
  - For each relevant category of constraint, characteristics of that constraint are also captured. For example, constraints for object persistence capture characteristics such as the number and size ranges of persistent objects, typical time period over which the object must be kept, the frequency of updates, and survival across hardware or software crashes.

Supplementary Specification

- Identify and characterize any restrictions or constraints in the form of non-functional needs or design constraints.
  - Design constraints can also include required sequences of operations.

Glossary

- Continue adding to the Glossary as part of developing the stakeholder needs.

Gather an Understanding of Architecture and Design

In the Inception Phase, the focus for this **gather** activity is twofold. One emphasis is on identifying and determining the architectural and design constraints imposed on the candidate solutions by any infrastructure on which the solution will run and any other systems with which the solution must interact. The overall context in which the new solution must operate must be captured. This context includes legacy capabilities with which the new solution shares data.

The second focus is on identifying generic architectural patterns that might be applicable. These architectural patterns will be investigated for use in **refine** when all **gather** activities are analyzed to formulate high-level architectures for the candidate solutions.

Determine architectural context

Architecture

- Understand the architectural context for the solution in sufficient detail to ensure that each candidate solution developed in response to the identified stakeholder needs will operate in the context of the larger organization. Capture this information in the Design Model.

- ❖ The Design Model will be the major blueprint for implementation of the solution. It captures the results of analysis and design into a single model.

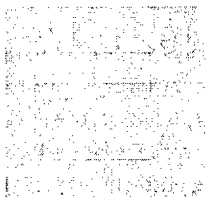
External interfaces

- ❑ Identify architecture constraints, boundaries, and interfaces posed by external systems with which the solution must interface. Capture this information in the Design Model and the Architecture Document.
  - ❖ The Current and Target Business Use-case Models should be a source of the specific external systems from which interfaces must be determined.
  - ❖ The Architecture Document will contain various high-level architectural views of the system, key decisions, and lessons learned.

Identify architectural alternatives

Alternative architectures

- ❑ Formulate potential architectural alternatives for further exploration and analysis during **refine** activities.
- ❑ Identify architectural or design techniques associated with potential architectural alternatives.



Gather an Understanding of Marketplace and Other Sources

An understanding of the characteristics of the relevant segments of the marketplace is essential. This includes understanding the other customers in the identified segment and how they are using related technologies and components. Additional information comes from examination of the components that may play a role in the new solution and the vendors who sell them or the agency that developed them.

The objective of this activity is to describe the components and characterize the motivation and direction of the sources of the components that may be applicable to the solution. The task that leads to this information is commonly termed "product evaluation," but it entails far more than examination of component features. Also included are tasks that examine the suppliers' health, supply chains, long-term strategies, business forecasts, and related items–typically characterized as market research. Preliminary information regarding architectural and design implications of the available components is also gathered, perhaps through limited demonstration of the components.

Identify relevant component sources

Look for any market trends that may affect either the architecture or the end-user business processes in the solution. Consider vendors' long-term support of the components used in

the relevant market segments (technology maturity, component obsolescence, component splits or mergers, vendors' going out of business, buy-outs, etc.). In addition, look for components that may drive the definition of the solution. Identify and work with other customers to gain leverage over vendors in defining and delivering the changes, patches, and components needed for the solution.

Market

- Identify market segments solving similar problems and requiring similar capabilities to those needed for this solution.
  - It is important to know who is solving similar problems and the rough shape of their solutions. In addition, it is important to know the forecast of both their needs and the components that support them.
- Determine and capture the size and distribution of each relevant market segment in the Market Segment Information.
  - For each technology area in the market segment, include the number of sellers (vendors/suppliers), approximate number of buyers, and the relative market share for each seller.

Candidate technologies and components

- Identify candidate technologies that are applicable and the leading components in each technology area.

Vendors/suppliers

- Characterize the vendors and suppliers in the relevant market segments.
  - Data on the behavior of the vendor/supplier market segment includes information on how vendors/suppliers differentiate themselves from each other, what competitive forces drive this market segment, what typical relationships vendors/suppliers have with their buyers, and what types of contracts and licenses are common.

Other customers

- Identify organizations that have similar needs for components.
  - Find out who they are and how much of the customer base they represent.

- Characterize the behavior of similar organizations in terms of their needs, their end-user business processes, and their use of available technologies and components.
  - ❖ Find out who they are, how much of the market they represent, how they use the components, and how they influence the market customer base to support their needs.

Standards

- Identify any applicable standards from the broader organization, government, or appropriate standards bodies.
- Identify any applicable commercial standards.
- Prioritize the standards applicable to this solution.

Characterize available components

In this phase, identify the components that are likely to support the solution. These components will come from a variety of sources and the published information about them will differ widely. It may, therefore, be necessary to bring demonstration versions of the most promising components into an experimental facility where the capabilities and limitations of each component can be determined. At this point, it is not important to learn as much as possible about each component (that will be done in the next phase) but to understand, in general, how components under consideration differ in how they approach the critical Use Cases.

Component screening

- Identify criteria for screening components. Capture in the Component Screening Criteria and Rationale.
  - ❖ As the project proceeds, new or changed components will be introduced. These components must be screened based on current screening criteria.
  - ❖ Early on, the criteria contain primarily basic component capabilities, vendor/supplier viability, and component scalability. As understanding grows regarding the stakeholder's needs and the components, the criteria evolve to include criteria used by previous **refines** to eliminate other components from consideration.
- Screen components against Component Screening Criteria and Rationale, capturing the rationale for removing any components from further consideration in the Component Screening Criteria and Rationale and the Component Dossier for that component.

Component capabilities

- Characterize (at a high level) the capabilities and limitations of candidate components in the Component Dossier with emphasis on the features that distinguish the components.
- Identify architectural or design assumptions made by components that reflect on the defined critical use cases (if any are defined at this point). Capture the assumptions in the Component Dossier.
  - For example, does the component presume or imply a centralized or distributed user base?

Vendor/supplier information

- Capture in the Component Dossier the component's vendor/supplier's market strategy, general release frequency, typical relationships with buyers, typical licensing arrangements, and long-term viability information.

Gather an Understanding of the Programmatics and Risks

Included in this **gather** activity is the management information that must be captured and monitored to define the solution and support the tradeoffs in the **refine** activity. This information consists of cost, schedule, and risk. Of particular note, (because they are often forgotten) are the costs, schedule, and risks associated with implementing the business process changes that are driven by the components. Also included is any management information (policies, constraints, etc.) from outside of this project that may affect the definition of the solution. This information will be gathered originally based on information developed external to the project and will be updated in further iterations to include project-specific information.

The identification of risks is really a pervasive task across EPIC. Risks can, and will, be identified at any point in the process and they should be documented, as described in the Risk Management Plan, as they are discovered. This **gather** activity provides a formal, systematic way to identify risks that may not show up in any single activity.

Identify management information

Cost

- Identify any cost constraints associated with the project. Determine how much room exists for negotiation.

Schedule

- Identify any schedule constraints for the project. Understand what drives these constraints.

Personnel
- ❑ Conduct an analysis of the gap between the skills, training, and capabilities of the team and those needed for this project.

Other constraints
- ❑ Identify and track any organizational policies that may apply to, or constrain, this solution.

Identify procurement needs and opportunities

Contract vehicles
- ❑ Identify and characterize available contracts and procurement vehicles for components and services.
  - ❖ Services may be delivered by third-party vendors who provide tailoring or integration.

Licensing
- ❑ Identify any available License Agreements for candidate COTS components.
  - ❖ Enterprise licenses are one example; but other parts of the organization may already have licenses that could be used for this solution.

Identify implications of potential changes to the end user's business process

Past technology insertion efforts
- ❑ Collect information on the previous process change history of the target organization.
  - ❖ Types of information include the frequency and nature of technology and changes to end user's business process over the past 5-10 years, success or failure of the changes, and the factors that led to the outcome.

Organizational change readiness assessment
- ❑ Collect data about the culture of the target organizations.
- ❑ Identify points of leverage to induce change.
- ❑ Identify sources of resistance.

Identify risks

Risks

- Update technical and non-technical risks in the Risk List.
  - It is important to have a consistent way to capture risk statements. Capture both the risk and the impact or consequence of the risk.
  - Consider complexity of the enterprise, the business domain to be explored, and any external constraints (e.g., cost, schedule, policy).
  - There will be different risks for each candidate solution.

Refine the Understanding of the Solution

Information has been **gathered** (and continues to be **gathered**) within each sphere of influence. The **refine** tasks for this iteration manage the interaction among and integration of the information from the **gather** tasks. This is accomplished by looking for patterns in how the information can be combined to form *candidate* solutions that meet the demands of the most critical use cases. In this phase, multiple solutions will be formed based on the gathered information; these solutions will be considered "candidates" until one is selected in the Elaboration Phase.

The emphasis in **refine** is to identify and resolve mismatches across the information through analyzing the major high-level characteristics and implications to the solution. As information about candidate solutions is analyzed, it may be found that additional information must be **gathered.** This will occur both as information is found to be incomplete and as the project team is ready to define the candidate solutions to an additional level of detail. Similarly, analysis of the information will expose mismatches that must be resolved through negotiation.

During these tasks, an understanding of the behavior of the marketplace and other sources drives the definition of candidate solutions. As more is learned about the capabilities and limitations of available components, some components may be dropped from consideration, some stakeholder requests and/or end-user business processes may be modified, some aspects of the architecture may be modified, or some candidate solutions may be dropped entirely.

Refine determines when candidate solutions are *sufficiently* harmonized, based on iteration objectives, to be assembled into an Executable Representation. In this phase, *each* candidate solution must be refined to the following point: the Solution Vision is articulated; the top-level design considerations are understood; and implications for the end-user business process are identified with tentative stakeholder agreement to implement the changes. These

understandings will be documented in a set of artifacts for each candidate solution (or in one set of artifacts with annotated variations if the differences are minor).

Each candidate solution will represent an alternative that delivers a useful capability, in the context of the critical use cases for this solution, within known constraints (technical and programmatic). The differences between these candidate solutions may be very small. In this instance, they may be considered as variations within a single solution. When the differences between the candidate solutions are large, they must be defined as separate, and independent, solutions. For example, separate solutions are required when the solutions differ significantly in their coverage of the business model or in their architectural implications.

Refine is not composed of sequential tasks. Rather, the tasks listed below represent work that is occurring concurrently. All tasks are both dependent on and critical to the other tasks with continuous feedback among them.

Synthesize information in candidate solutions

The following steps provide a "quick look," pair-wise comparison of the identified components to the information gathered from "stakeholder needs and business processes" and "architecture and design." The focus is on trying to understand where they match, where they don't match, and a characterization of the mismatch.

With a basic understanding of the matches and mismatches, high-level candidate solutions can be formed (or updated) through negotiation with the affected stakeholders. The organizing principle for the candidate solutions is the architecture that defines the major elements of the solution and the linkage between them. The architecture for the solution must be flexible enough to accommodate the current component(s) and the projected growth path for those components to optimize the ability to evolve the solution efficiently as components and technologies change.

The steps below describe the basic steps that must be completed, but the order is subject to the needs of a particular iteration; they will seldom be implemented in the order shown. In most cases, cycling between the steps is required as resolution of mismatches in one sphere introduces changes to the baseline and, therefore, potential new mismatches in another sphere.

End-user business processes

- Determine, at a high level, how the components will affect business processes across the broader organization.
- Compare the components to the target end-user business processes.
- Identify potential mismatches in the end-user business process. Characterize the nature of each mismatch.

Critical Use Cases

- Identify the coverage of components against the critical use cases. Determine where the components appear to overlap, where the sum of the components appears to be deficient, and where the components appear to provide functionality that is not currently requested.

Use-case mechanisms

- Identify how the components support the mechanisms and services that are needed by the critical Use Cases. Characterize any mismatches.
  - ❖ Mechanisms/services include any needs for persistence of data, fault management, messaging, error management, and transaction management.

Non-functional

- Identify how components support the quality attributes and services needed in the Supplementary Specification. Characterize any mismatches.

Architecture

- Identify, at a high level, where components appear to fit and where there are potential mismatches with the broader organizational environment (including any existing external interfaces).
  - ❖ Look at existing networks (e.g., protocols), databases, database designs, firewalls, servers, and naming standards.

Candidate solutions

- Determine how combinations of components could work together to form candidate solutions that minimize the high-level mismatches. Identify where custom components might be needed. Capture this information in the Design Model.
  - ❖ The mismatches from the pair-wise comparisons form the basis for this step. Each mismatch is either resolved in forming a candidate solution, or becomes a mismatch carried forward with the candidate solution for further analysis.
  - ❖ You need to know if "glue" will be needed (and if so, a rough order of magnitude of how much) and what kind of component integration strategy might link the components.
  - ❖ Limited prototypes using actual components may be required for this step.

- Characterize the end-user business process implications and other mismatches for each candidate solution.

Cost and schedule

- Estimate the rough costs and schedule associated with building, fielding, and supporting the candidate solution.
  - These estimates are intended only to determine whether there are large discrepancies between the candidate solution and the constraints–they will be reworked in subsequent tasks.
  - Include the cost and schedule for implementing any agreed-upon changes to the business process.
  - Total cost of ownership must include component upgrades and the cost for any reintegration, retesting, and refielding of the solution throughout the expected life of the solution.
- Identify where the candidate solution exceeds any known cost or schedule constraints (mismatches).

Screened components

- Components that are not included in any candidate solution will be screened from further consideration. Capture the rationale for removing any components from consideration in the Component Screening Criteria and Rationale.
  - The rationale for removing components amplifies the screening criteria for similar new components identified in subsequent **gather** activities.

Analyze and negotiate mismatches for each candidate solution

The focus in this task is on understanding and resolving, if possible, the mismatches previously identified in each candidate solution. It is important to understand how important each mismatch is to the solution and understand the possible ways the mismatch can be resolved. Mismatches can be resolved through negotiating with stakeholders to modify end-user business processes and stated stakeholder needs; gathering more information about the capabilities of the components and their ability to be tailored to accommodate the mismatch; changing the way the architecture uses the components; or creating a custom component to provide the necessary capability. If a mismatch cannot be sufficiently resolved, the candidate solution may be removed from further consideration.

The steps below describe the basic steps that must be completed, but the order will be subject to the needs of a particular iteration; they will seldom be implemented in the order shown. In most cases, cycling between the steps will be required as resolution of

mismatches in one sphere introduces changes to the baseline and, therefore, potential new mismatches in another sphere.

To capture the results of this task, candidate solution-unique addendums will be created for the Target Business Use-case Model, Target Business Object Model, Use-case Model, critical Use Cases, the Supplementary Specification, and the Design Model. These addendums will persist as long as the candidate remains under consideration. In addition, the Business Case will eventually contain summary information about each candidate solution considered. It is introduced here to capture information about any candidate solutions removed from consideration.

Critical Use Cases

- Identify mismatches where the candidate solution falls short in meeting the needs of the critical Use Cases. Determine the impact of not meeting this need.
  - The Use Cases include both functional behaviors and the quality attributes (like performance) specific to a use case.
  - It may be useful to understand how other customers with similar needs address these shortfalls.
- Identify mismatches where the candidate solution operates differently from the behavior required in the critical Use Cases. Determine the impact of changing end-user business processes to match the solution.
- Identify mismatches where the candidate solution provides functionality not addressed in the Use-case Model. Determine if the additional features adversely affect end-user business processes or, perhaps, offer an opportunity to optimize end-user business processes.
- Resolve mismatches, where possible, through negotiations with the affected stakeholders. Record the results of the negotiation in the Target Business Use-case Model, the Target Business Object Model, the Use-case Model, and the Supplementary Specifications as appropriate.
  - Consider techniques such as Win-Win Requirements Negotiation [17, 18], or any of the techniques described in *Getting to Yes* [19].
- Identify any custom components necessary to resolve mismatches that cannot be negotiated.

Non-functional

- Identify mismatches where the candidate solution falls short in meeting the needs identified in the use-case mechanisms and the Supplementary Specification. Determine the impact of not meeting this need.
  - ❖ The Supplementary Specification includes the quality attributes of the solution that are not specific to a use case.
- Identify mismatches where the candidate solution operates differently from the behavior required in the use-case mechanisms or the Supplementary Specification. Determine the impact of changing end-user business processes to match the solution.
- Identify mismatches where the candidate solution provides quality attributes not addressed in the use-case mechanisms or the Supplementary Specification. Determine if the additional attributes adversely affect end-user business processes or, perhaps, offer an opportunity to optimize end-user business processes.
  - ❖ An example might be a component that makes security provisions that had not been considered in the original stakeholder needs.
- Resolve mismatches, where possible, through negotiations with the affected stakeholders. Record the results of the negotiation in the Use-case Model and Supplementary Specifications.
  - ❖ It may be possible to narrow the application of a particular quality attribute (e.g., the attribute is needed only under particular situations or in particular parts of the solution).
- Identify any custom components necessary to resolve mismatches that cannot be negotiated.

Architecture

- Identify mismatches where components within the candidate solution do not have needed interfaces or behaviors to link effectively into the architecture, with each other, with existing infrastructure, or to the broader organization's architecture. Determine the impact of not meeting these interfaces or behaviors.

- Identify mismatches where components overlap. Determine the ease with which certain functionality can be bypassed.

- Identify mismatches where the components provide interfaces not addressed in the architecture. Determine if the additional interfaces adversely affect the architecture or, perhaps, offer an opportunity to optimize the architecture.

- Resolve mismatches, where possible, through negotiations with the affected stakeholders. Record the results of the negotiation in the Design Model.

  - In this case, the affected stakeholders are likely to be vendors, other suppliers, architects, senior designers, and infrastructure owners.

- Identify any custom components, wrappers, or integration mechanisms ("glue") necessary to resolve mismatches that cannot be negotiated.

  - You need to know if "glue" will be needed (and if so, a rough order of magnitude of how much) and what kind of component integration strategy might link the components.

Cost and schedule

- Resolve any mismatches among cost, schedule, and the capabilities in the candidate solution through negotiation with the affected stakeholders. Record the results of the negotiation in the Project Plan, the Use-case Model, the Supplementary Specifications, and/or the Design Model as appropriate

  - The resolution to mismatches with cost or schedule includes relaxing the cost or schedule constraint or removing capability from the solution.

Screen candidate solutions

- Any candidate solutions with mismatches that could not be negotiated should be considered for removal from further consideration. Capture the rationale for removing any candidate solutions in the Business Case.

  - The focus here is showstoppers. In this phase, very little may be known about the candidate solutions or the included components. Some mismatches may appropriately be deferred for further analysis in later iterations or the Elaboration Phase.

Characterize each candidate solution

Because each candidate solution could contain very different combinations of components, and the components address different aspects of the critical Use Cases, the definition of what the solution will do and how it will do it may be different. The general content, high-level architecture, cost, schedule, risk and some management or technical plans must be documented for each remaining candidate solution. The Solution Vision captures *what* the solution will do. The architecture describes *how* the Solution Vision will be implemented. Because of the potential differences between candidate solutions, there will be candidate-unique aspects of the Solution Vision, Architecture Document, Target Business Use Case, Target Business Object Model, Test Plan, Deployment Plan, Development Plan, Risk List, and Project Plan. How these unique aspects are represented in the actual artifacts will depend on how much variability exists among the candidate solutions and the needs of the project (specified in the Development Case).

Solution Vision

- Amplify, or create, a Solution Vision to identify the major features (including growth vectors, quality attributes, and priorities) associated with each candidate solution.
  - The Solution Vision is the high-level customer view of the solution describing what the solution will do to solve the most critical problems and needs.
  - Features should relate "must have" needs, critical Use Cases, Supplementary Specifications, and the capabilities of the components for the solution. Features can be either functional or non-functional.
  - As a guide, the preferred number of high-level features is less than 100, with an ideal between 25 and 50.

Architecture Document

- Capture the salient attributes of the components, critical Use Cases, and Design Model for each candidate solution in the Architecture Document.
  - Capture information about the combination of components, component integration strategy, any custom code elements, unresolved mismatches, and the resolution and rationale for any negotiations and tradeoffs for the mismatches.
  - The Architecture Document will contain various high-level architectural views of the system, key decisions, and lessons learned.

- ❖ The architecture must accommodate the critical use cases, candidate components, the broader organization's architecture, applicable architecture standards, and any identified design constraints.

- ❖ The architecture should address the critical non-functional needs as well as the functional needs.

End-user business process change impacts

- ❑ Identify the delta (width and depth) from current to target end-user business processes represented in each candidate solution. Capture in the Business Process Change Management Plan.

- ❑ Identify a migration strategy and develop a high-level plan that migrates the affected organizations from the current to the target end-user business processes for each candidate solution.

  - ❖ This plan is still intended to be very high level. The emphasis here will be on capturing the aspects of the plan that could drive cost or schedule for the candidate solution.

Implementation

- ❑ Capture any attributes associated with building, fielding, or supporting each candidate solution that will drive cost or schedule in an addendum to the Development Plan, Test Plan, and Deployment Plan as appropriate.

  - ❖ Special consideration should be given to implications to test and fielding.

Vendor/Supplier Relationship Plan

- ❑ Identify and categorize the major vendors/suppliers of components in each candidate solution based on their importance to the solution.

- ❑ Develop a strategy for maintaining suitable information channels and for influencing component directions appropriate to the importance of each major vendor/supplier for each candidate solution.

- ❑ Capture any unique needs for contracts to support each candidate solution through the life of the solution in the Vendor/Supplier Relationship Plan.

Waivers

- ❑ Identify any waivers necessary for "compliance" with the broader organization's architecture or applicable policies for each candidate solution.

Risks

- Update the Risk List for each candidate solution. Analyze and prioritize the identified risks for each candidate solution.

Cost and Schedule

- Refine the estimate of the cost of building, fielding, and supporting each candidate solution. Capture this information in the Business Case.
  - Include the cost for implementing any agreed-upon business process changes.
  - Total cost of ownership must include component upgrades and the cost for any re-integration, re-testing, and re-fielding of the solution throughout the expected life of the solution.
- Refine the estimate of the schedule for building, fielding, and supporting each candidate solution. Capture this information in the Business Case.
  - This schedule information is unique to each candidate solution. The planning activity in the next iteration plan will have to accommodate the important variations.
  - Include the schedule for implementing any end-user business processes–this may take longer than integrating components.

Business Case

- Capture, for each remaining candidate solution, the general functionality, performance, quality, fielding approach, changes to the end user's business process, cost/benefit, schedule, and risks over the anticipated life of the solution.

Screen candidate solutions

- If necessary, screen candidate solutions to a reasonable number for further consideration.
  - It is important that the Business Case shows that all possible alternatives were explored. Yet, it is too expensive to explore a large number of candidates in Executable Representations and further iterations if there are a reasonable number of high-value candidate solutions and a credible way to differentiate between them.
- Record the rationale for eliminating any solution in the Business Case.

Assemble an Executable Representation

The effort to assemble one or more Executable Representations begins when the candidate solutions have reached sufficient fidelity to meet the iteration objectives. *Assemble* includes configuring and testing the necessary infrastructure and components of the candidate solutions.

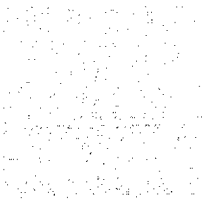
Build and test proof of concept(s)

Implementation Model	❑ Define the organization of the Executable Representation in terms of needed components.
Test Set Artifacts	❑ Develop test scripts and other Test Set Artifacts needed to evaluate the Executable Representation.
Implement and test components	❑ Write any source code needed, adapt existing components, compile, link, test, and execute.
	❑ Tailor and test the components.
	❑ Develop and test integration code or data (wrappers, glue, data sets, etc.) needed to incorporate pre-existing and custom components.
Integrate and test	❑ Integrate new and changed components into a new version.
	❑ Execute appropriate integration and solution level tests.

Prototype the needed changes to the end user's business process

As the proof of concept is being assembled, the end users must prototype the changes to their business process associated with the candidate solutions.

Change management	❑ Prototype needed changes to the end user's business process.
	❑ Prototype the appropriate elements of the Business Process Change Management Plan.



Assess the Iteration

As the iteration completes, it is important to determine if the objectives planned for this iteration were achieved (any unmet objectives will be assigned to future iterations). In addition, a review of any unplanned questions, risks, or issues that arose during the iteration must be conducted so that they can be captured in the appropriate planning artifacts.

Assess the Executable Representation

The Executable Representation provides an opportunity for both the end users and the engineers to evaluate the evolving candidate solutions in the context of the objectives for the iteration. End users must verify that the end-user business processes represented are acceptable. Engineers must verify that each candidate solution can be implemented. Together they verify that the iteration's objectives have been met and that the evolving candidate solutions meet real needs, that the end-user business processes prescribed by the solution are acceptable, and that the solution can be implemented.

- Validate the solution.
  - Show that the solution is what the stakeholders need or want.
- Verify that the solution is implemented correctly.

Update the information about the solution

Screen candidate solutions

- Define and update the criteria that will be used to select a single solution.
- If there is enough information, screen candidate solutions to select a reasonable number to be examined in detail in the Elaboration Phase.
  - Solutions may be screened in any iteration as information is discovered that shows that one or more solutions is not suitable. Once a single solution has been selected, iterations will skip this activity.

Business Case

- Amplify the Business Case to capture the significant decisions made in the iteration. In particular, the rationale for selection of a single solution should be captured.

Determine lessons learned from iteration

Iteration Assessment

- ❑ Determine if the objectives planned for this iteration were achieved (selected risks mitigated).
  - ❖ Unmet objectives will be reassigned to future iterations.
- ❑ Identify any unplanned questions, risks, or issues that arose during the iteration and assign them to future iterations.

Risk List

- ❑ Update the Risk List based on the Iteration Assessment.

Project process improvement

- ❑ Review project metrics and make recommendations for process improvement.

Assess the phase, if the iteration completes the phase

Assessment group

- ❑ Appoint an assessment group with representatives for all affected stakeholders, including end users.

Phase exit criteria

- ❑ The assessment group determines whether the phase objectives and exit criteria have been met and decides whether the project should go ahead. This constitutes the LCO anchor point.
  - ❖ The phase exit criteria should have been documented in the Project Plan. Much of the necessary information that shows whether or not those criteria are met should have been captured in the Business Case.
- ❑ The assessment results are captured in the phase Review Record.

Supporting Activities

The activities associated with project monitoring and control, technical process activities, and supporting process activities that were included in the Development Plan have not been described in the activities laid out in this document. They are, however, critical to the success of the project. Use of the Rational Unified Process, or any comparable rigorous process description, in these areas is necessary to the EPIC process.

Monitor project status

- Monitor the progress of the project and quality of the solution relative to the Project Plan (including budget and schedule) and from the viewpoints of the various stakeholders.
  - This includes the monitoring metrics that measure the progress of the solution.
  - Capture and assess the measurements associated with the project's business measurement goals. Verify that the right goals are being measured.
- Discover exceptions and problems that must be resolved for project success.

Prepare experimentation facility

Prepare the experimentation facility in accordance with the Infrastructure Plan to support the identified tasks for this iteration.

Update and create contracting vehicles as necessary

Appropriate procurement vehicles must be in place to support the ongoing activities in the Inception Phase and the projected activities in the Elaboration, Construction, and Transition Phases.

Contracts and procurement vehicles	❑ Update and create any contracts and procurement vehicles for needed components and services.
License Agreements	❑ Negotiate any licenses needed to support examination of the components.

Phase Artifacts

The artifacts for the Inception Phase are intended to capture the agreed-upon solution scope and to verify that at least one candidate solution is feasible. Most of the artifacts are started in outline form and will be expanded across later phases as more information is gathered and refined.

TO CHARACTERIZE END-USER BUSINESS PROCESSES AND STAKEHOLDER NEEDS

Current Business Use-case Model
Current Business Object Model
Target Business Use-case Model
Target Business Object Model
Glossary
Stakeholder Requests
Use-case Model
Use Cases
Supplementary Specification

TO CHARACTERIZE THE MARKETPLACE AND OTHER SOURCES

Market Segment Information
Component Dossier (for each examined component)
Component Screening Criteria and Rationale

TO CHARACTERIZE THE ARCHITECTURE AND DESIGN

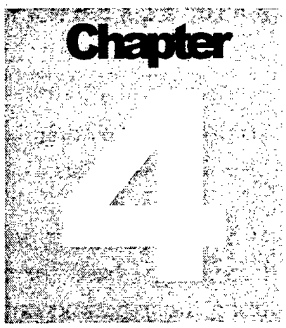
Solution Vision
Architecture Document

Design Model
Executable Representation(s)
▪ Implementation Model
Test Set Artifacts (includes the Test Plan)

TO CHARACTERIZE PROGRAMMATICS AND RISK

Development Plan
Management Process
▪ Project Plan
▪ Iteration Plans
Project Monitoring and Control
▪ Requirements Management Plan
▪ Schedule Control Plan
▪ Budget Control Plan
▪ Quality Control Plan
▪ Reporting Plan
▪ Measurement Plan
Risk Management
▪ Risk Management Plan
Technical Process Plans
▪ Development Case
▪ Infrastructure Plan
▪ Solution Acceptance Plan

Supporting Process Plans
▪ Configuration Management Plan
▪ Evaluation Plan
▪ Documentation Plan
▪ Quality Assurance Plan
▪ Problem Resolution Plan
▪ Vendor/Supplier Relationship Plan
▪ Process Improvement Plan
Deployment Plan
Business Process Change Management Plan
Business Case (includes business context, success criteria, financial forecast)
Risk List
License Agreements
Iteration Assessments
Review Record



Elaboration Phase

The goal of the Elaboration Phase is to achieve sufficient stability of the architecture and requirements; to select and acquire components; and to mitigate risks so that a single, high-fidelity solution can be identified with predictable cost and schedule.

Phase Description

The Elaboration Phase establishes a common, stable baseline among the affected stakeholders for the one solution that will be built and fielded. This baseline includes the framework and strategy for implementing the solution. It includes the definition of any required business process changes; the architecture that links the pre-existing and any required custom components; and the implications (including cost, schedule, and associated risk) of implementing them. It also includes requirements that describe the functional and non-functional capabilities that must be in the fielded solution (i.e., establish the minimum success criteria for the solution). Requirements are formed based on negotiations between a detailed understanding of what the components can provide and the needs and desires of the various stakeholders.

LCO marks a change in intensity. The basic activities for the Elaboration Phase are the same as those in the Inception Phase, but the level of detail is deeper and the level of resource commitment is significantly higher. The focus of the Elaboration Phase is on in-depth hands-on experiments with the candidate solutions by end users and engineers. These experiments and prototypes are conducted in an experimentation facility that represents, as closely as practical, the operational environment.

This phase includes further definition of stakeholder needs and end-user business processes based on detailed evaluation of the components. Software design and implementation activities are expected during the Elaboration Phase. These are aimed at definition and prototyping of the strategy and mechanisms for component integration, data migration, and

component tailoring.[14] The focus continues to be keeping the four spheres in balance as greater knowledge of each of the candidate solutions is gained.

When the candidate solutions are sufficiently understood, one solution is selected that will become the basis for the Construction Phase. The selected solution is further amplified, using the experimentation facility, until it is shown that the selected solution has achieved sufficient stability in requirements and architecture as demonstrated in an Executable Representation.

The objective of the experiments in the Elaboration Phase is to better understand how the components contained in each candidate solution affect the end-user business processes and stakeholder needs as well as to define and validate an architecture that links the pre-existing components, any custom components, and the interfaces to the broader organization. Experimentation also allows informed trades among the spheres of influence to be made. Where possible, end-user business processes are modified and stakeholder needs negotiated to allow greater use and leverage of available components.

As the components are explored, the cost of each candidate solution is estimated. These cost estimates must include not only the purchase fees for each component, but also the cost to continue monitoring the marketplace, evaluating new components, and incorporating new releases of the components through the life of the solution. The cost to incorporate new releases must include the costs to re-integrate, re-test, and re-field the solution.

The results of all tradeoffs and stakeholder negotiations are documented to form a documented baseline of the solution for use in the Construction Phase. The principal artifacts of this phase are

- the Executable Representation in the form of an evolutionary prototype of the solution's architecture
- the Solution Vision that summarizes both the problem and the solution at a high level of abstraction by capturing the main characteristics, major features, key stakeholders needs, and key services to be provided
- the Solution Requirements Specification that describes the negotiated functional and non-functional requirements that must be included in the solution (contains the Use Cases and the Supplementary Specification)
- the Design Model that is the major blueprint for the implementation of the solution. The model consists of a set of collaborations of classes, packages, and subsystems that provide the behavior of the system.
- the Architecture Document that provides a comprehensive architectural overview of the solution, using a number of architectural views to depict different aspects of the solution, and that captures the architecturally significant tradeoffs and decisions

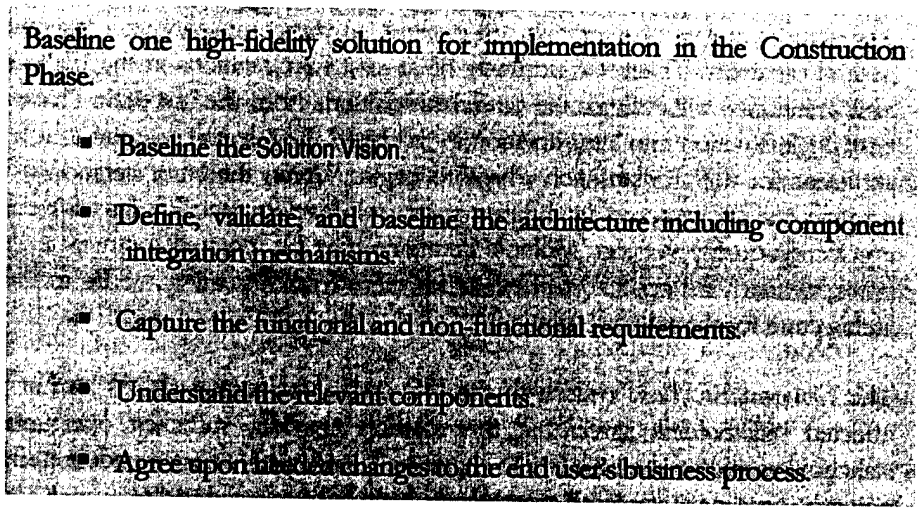
[14] *Tailored* means non-source code adjustment necessary to integrate the COTS products into an operational system (e.g., scripts).

- the Business Process Change Management Plan that describes how the Target Business Use-case Model will be implemented by the affected end users

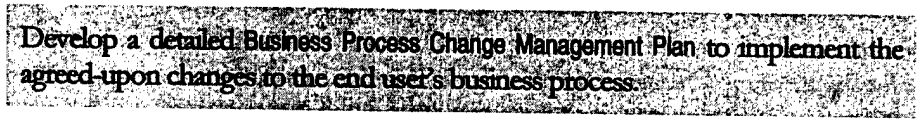
The actual number of iterations will depend on the complexity and risk inherent in the desired capability. There will normally be at least two iterations in the Elaboration Phase; early iteration(s) will evaluate the candidate solutions from the Inception Phase to select the best single solution and later iteration(s) will build the selected solution to a level of detail sufficient for the Construction Phase to begin. Within the latter iteration(s) the selected solution will be explored to mitigate risks and develop specific plans for constructing a production-quality version. Multiple iterations may be required to mitigate the technical, programmatic, and operational risks and achieve a stable baseline of the requirements and architecture for the selected solution.

The Elaboration Phase ends with the Life-cycle Architecture (LCA) anchor point when affected stakeholders agree that the solution provides sufficient operational value to stakeholders and can be built by the engineers for acceptable cost, schedule, and risk. At this point the baseline is stable; all components have been selected and procured; any integration mechanisms to incorporate the pre-existing components and any other components are validated; and the cost and schedule for building, fielding and supporting the solution have been predicted within an acceptable range. All significant risks are eliminated or acceptable risk mitigation plans are in place.

Phase Objectives



Demonstrate that the baseline solution (including components and their integration mechanisms) will support the Solution Vision at a reasonable cost in a reasonable time.



Acquire the components and services needed for the Construction Phase and initial fielding in the Transition Phase.

Phase Task Overview

Phase Planning Activities
- Update the Development Plan for the project

Iteration Activities

Plan the Iteration
- Build a detailed plan for the iteration
- Update the Development Plan for the project

Gather ...

... an Understanding of Stakeholder Needs and End-User Business Processes
- Update and expand the business model
- Capture the significant behaviors of the solution

... an Understanding of Architecture and Design
- Amplify the architectural context
- Amplify the architectural alternatives contained in solution(s)

... an Understanding of Marketplace and Other Sources
- Monitor relevant component sources
- Evaluate applicable components

... an Understanding of the Programmatics and Risks
- Update management information
- Update procurement needs and opportunities
- Amplify implications of potential changes to the end user's business process
- Update risks

Refine the Understanding of the Solution
- Identify and resolve mismatches from the synthesis of new information
- Amplify the solution(s)

Assemble an Executable Representation
- Build and test an architectural prototype
- Prototype the needed changes to the end user's business process

Assess the Iteration
- Assess the architectural prototype for the solution(s)
- Update the information about the solution
- Determine lessons learned from iteration
- Assess the phase if the iteration completes the phase

Supporting Activities
- Monitor project status
- Prepare experimentation facility
- Update and create contracting vehicles as necessary

Phase Exit Criteria

Status	Exit Criteria
	Affected stakeholders agree that within acceptable risk, the vision of one solution can be met, in the context of the solution architecture, if the current plan to build the solution is executed. ▪ The requirements are identified and negotiated to optimize leverage of the marketplace and other sources. ▪ The architecture is stable and reflects a structure that is flexible enough to support the continuing evolution of components and operational needs. ▪ The stakeholders, who are subject to the changes, support the necessary changes to the end user's business process. ▪ Cost/schedule estimates and project tasks are credible.
	An Executable Representation demonstrates the common understanding of the solution that has been achieved through negotiation with stakeholders and addresses (and resolves as appropriate) major risks of the solution.
	The remaining risks are understood, are acceptable and have appropriate mitigations identified.
	The Business Process Change Management Plan realistically accounts for all types of end users and the necessary changes for both individuals and their organizations.
	The project has defined and implemented relationships with key vendors/suppliers that provide needed insights into component capabilities and directions. (In the Vendor/Supplier Relationship Plan)
	Any differences between actual resource expenditures versus planned expenditures for this phase are understood, and corrective actions are included in the Project Plan.
	The experimentation facility and the Development Case are sufficient to support beginning the Construction Phase.
	The Development Plan has been updated. (The plan for the Construction Phase is sufficiently detailed and accurate and is backed up with a credible basis for all estimates.)
	The License Agreements, contracts, and procurement vehicles for needed components and services are in place for the Construction Phase and initial fielding.

Phase Planning Activities

Update the Development Plan for the project

The Development Plan is updated in each iteration based on new information and lessons learned in engineering and management activities in previous iterations. The Development Plan references the entire set of major planning artifacts that describe the project and how it will be executed. In this phase the plans necessary to implement the solution in the Construction Phase are being developed in parallel to the evolving definition of the solution. Simultaneously, more is understood, and captured, about what will be required to field the solution in the Transition Phase. By the end of the Elaboration Phase, a detailed plan for the Construction Phase is prepared and the Development Plan is updated accordingly.

Development Plan

- Review the project organization. Verify that the level of staff resources and skills available are appropriate.
- Update the Project Plan.
- Review the plans for project monitoring and control for updates needed.
- Update the Risk Management Plan.
- Complete and maintain the Development Case.
  - Evaluate the skills mix, roles and responsibilities, and reporting structures of the current engineering organization.
  - Assess the current tools support and select tools for use for engineering and integration.
  - Produce project-specific templates for key artifacts.
- Review the remaining plans for technical process planning and update as required. In particular, update the planning for the experimentation facility in the Infrastructure Plan.
- Update and categorize the major vendors/suppliers based on their importance to the project. Capture these in the Vendor/Supplier Relationship Plan.
- Update the strategy for influencing/maintaining information channels appropriate to the importance of each major vendor/supplier in the Vendor/Supplier Relationship Plan.

- Review the plans for the other supporting processes (such as configuration management and quality assurance) and update as needed.

Test planning

- Amplify the Solution Acceptance Plan to document all of the criteria for stakeholder acceptance of the delivered solution.
  - ❖ It is critical to have stakeholders validate that the acceptance criteria are complete.
- Update the test objectives and strategy in the Test Set Artifacts.

Plan for fielding

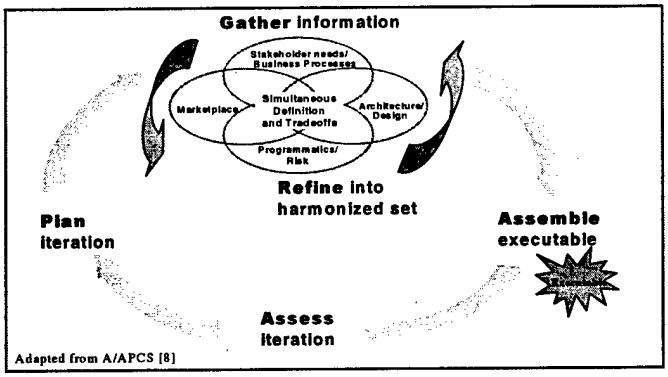
- Update and validate the Deployment Plan with affected stakeholders; include plans for data migration.
  - ❖ Installation of components may require a level of technical expertise that is not currently available in the field. Planning for the solution must accommodate this shortfall.
  - ❖ Bear in mind that each site's implementation may be somewhat different. The Deployment Plan must take these differences into account and explicitly allocate responsibilities for handling site-unique tasks.
  - ❖ For fielding capabilities that have a high risk of loss of data or functionality, consider parallel operation with the legacy system that the solution is replacing.

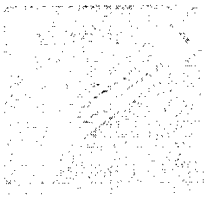
End-user business process change planning

- Update and validate with end users a detailed Business Process Change Management Plan, including a resistance mitigation strategy, for needed changes to organizational structure and/or end-user business processes to match the Deployment Plan.
  - ❖ The Organizational Change Readiness Assessment in the Business Process Change Management Plan provides key inputs to this plan.

Iteration Activities

ITERATION ACTIVITIES KEY
Plan
Gather
Refine
Assemble
Assess





Plan the Iteration

Plan the iteration in detail at the beginning of each iteration. The known risks captured in the Risk List are a key consideration for defining the iteration objectives and activities. The information (and the level of detail of that information) to be **gathered** about each of the spheres of influence is determined by the iteration objectives. The iteration objectives also determine what analysis must be done to **refine** the information from each of the spheres. Any lessons learned and new risks discovered from previous iterations are incorporated.

Build a detailed plan for the iteration

The detailed plan for the iteration is based on the current Risk List and the most critical functionality. Risk is a key discriminator in deriving objectives for the iteration. The highest priority risks are mitigated as early as possible. However, addressing risk is balanced with ensuring that the critical functions and services that a solution must provide are addressed early.

Iteration Plan

- Refine the scope of the iteration and the goals and objectives that were planned in the Project Plan for this iteration to reflect any changes since the plan was last updated.
- Define objectives for the success of the iteration.
  - ❖ These objectives will provide focus for all activities in the iteration and will be used at the end of the iteration to decide whether the iteration was a success.

- Define the requirements for the Executable Representation needed to demonstrate the iteration objectives.
  - An Executable Representation prototypes the architecture and shows feasibility and consistency with the broader organization's architecture.
  - This prototype is in addition to one or more *exploratory*, throwaway prototypes to mitigate specific risks such as design and requirements tradeoffs, component feasibility analyses, or demonstrations of key scenarios to certain stakeholders conducted as part of **gather** or **refine**.
  - Depending on the specific objectives of the iteration, there may have to be more than one prototype.
- Identify the tasks that will be required to achieve the iteration objectives and the specific artifacts that must be developed or updated.
- Complete a detailed work breakdown structure to show how the work that must be done within the iteration is allocated and what resources are necessary to do the work.
- Determine milestones (events and dates) that are important to the iteration.

Update the Development Plan for the project

Development Plan

- Revise and update the Development Plan.
  - The plan should be updated to reflect changes to the technical or programmatic baseline, to reflect changes in personnel availability or skills, to reflect the changes necessary to accommodate a particular set of components, or simply to reflect a new approach to meeting the identified needs.

Gather Information

There is a **gather** activity, comprising discrete tasks, that collects the information needed to meet the iteration objectives from each of the four spheres of influence and that builds representations of the information specific to the type of information gathered.

The information gathered from one sphere depends on and drives the information needed from another sphere. Yet, it is useful to think about information from these spheres individually because the nature of the information in each sphere is fundamentally different, the dynamics of the information are different, and the techniques associated with gathering information from each of the spheres are different. In practice, this drives the process to gather a little, refine a little, then gather some more, then refine some more. The **gather** tasks within each of the four spheres, therefore, occur concurrently. In addition, the **refine** tasks for this iteration manage the interaction among, and integration of the information from, the **gather** tasks. The **gather** and **refine** activities will continue to cycle until the information becomes sufficiently detailed to meet iteration objectives.

In the Elaboration Phase, the emphasis in **gather** will be on

- monitoring the information gathered during Inception for changes
- adding details to the critical Use Cases
- adding any additional Use Cases that are "significant" in that they include functionality that will drive component and architecture decisions
- performing a detailed evaluation of the components under consideration

The iteration tasks described below apply to both early and late iterations; they will use *solution(s)* to represent both the candidate solutions (early iterations) and the selected solution (late iterations).

Gather an Understanding of Stakeholder Needs and End-User Business Processes

The emphasis in this phase is to build or improve the understanding of the detailed needs, boundaries, and constraints of the solution. This understanding will be reflected in expanding the Use-Case Model to include the significant Use Cases that drive the solution's functionality and major design tradeoffs and that are captured in the Stakeholder Requests that support those Use Cases.

Update and expand the business model

The business model was described, at least at a high level, in the Inception Phase in the Target Business Use-case Model and the Target Business Object Model. In the Elaboration Phase, capture and amplify any remaining end-user business processes that affect or are affected by the solution(s).

Mission

- Review and update the understanding of the mission and strategic direction of the end-user organization as well as the forecast of potential changes and forces of change that will affect the end-user organization and its mission over time.
- Update the business goals of the end-user organization that relate to, or are affected by, the solution.

Stakeholders

- Verify that all stakeholders with a vested interest in the outcome of the solution have been identified.

Current fielding capabilities

- Update and amplify the characterization of the current operational environment for each of the relevant stakeholders who will use some aspect of the solution. Understand their current capabilities and available resources for system upgrades or major enhancement.
  - The end user is able to incorporate certain kinds of changes easily while other changes are more difficult. This may affect the definition of the solution and will drive the planning for fielding. The culture of the organization will determine which changes are easy or difficult to implement.
- Verify that current fielding capabilities are understood as well as any methods and resources currently in place to implement system upgrades or major enhancements.

Change drivers

- Update the understanding of why each change in the current environment is thought to be necessary. To help characterize the issues that have to be considered, determine the root causes of the problems or shortcomings in the current business environment.

Business measures

- Review and update the measurable business improvement goals and the metrics that will be used to evaluate improvements from the solution(s) implementation.

Business modeling

- Review the objectives for modeling the current and target end-user business processes and the level of detail to which the modeling effort should go to meet those objectives.
- Review and expand the Current Business Use-case Model and Current Business Object Model as needed to support transition to the solution(s).

- Review and expand the Target Business Use-case Model and Target Business Object Model as necessary to reflect the implications of the solution(s).
- Continue adding to the Glossary.

Capture the significant behaviors of the solution

The purpose of this task is to amplify the information from the stakeholders regarding what their needs really are. Where in the Inception Phase the critical Use Cases were developed, in the Elaboration Phase Use Cases are developed for any behaviors considered either significant or critical; i.e., those cases that will drive component choices and other architectural decisions. The objective of the use-case modeling is to understand the required behavior of the solution (in contrast to the Business Use-case Model and Business Object Model that focus on the behavior of the business).

The Supplementary Specifications capture the non-functional behaviors and quality attributes of the solution that are not readily captured in the Use Cases (for example, legal and regulatory requirements and application standards; quality attributes of the solution, including usability, reliability, performance and supportability; and other requirements such as operating systems and environments, compatibility requirements, and design constraints).

Stakeholder Requests

- Review and amplify Stakeholder Requests (the raw input and "wish lists" from the various stakeholder groups) that relate to the solution(s).
- Challenge and sort stakeholder requests into "must have" needs.
  - There may be differences in stakeholder requests and needs for each candidate solution, as each candidate solution will have to be evaluated on its own merits.
- Amplify and prioritize important needs from stakeholders. These will serve as critical attributes to be used in prioritizing and bounding the requirements.

Use-case Model

- Review and update the identified set of Use Cases; validate with stakeholders that the right set has been captured; update the use cases as needed.
- Identify any additional remaining Use Cases and actors.
- Reprioritize Use Cases.

- Structure the Use-case Model looking for similarities, simplifications, and opportunities to improve the model. The goal is to make the model easier to understand and modify.

Critical Use Cases

- Provide detail for all critical Use Cases.

Significant Use Cases

- Develop descriptions for all significant Use Cases (i.e., those that drive the architectural, component, end-user business process, and planning decisions).

Use-case mechanisms

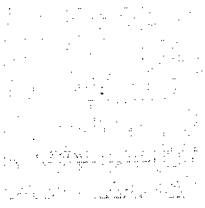
- Amplify the mechanisms and services that are needed by the critical Use Cases. Identify the mechanisms that are needed by the significant Use Cases.
  - Examples include data persistence, security features, distribution and concurrency, transaction management, and fault tolerance.
  - For each relevant category of constraint, characteristics of that constraint are also captured. For example, constraints for object persistence capture characteristics such as the number and size ranges of persistent objects, the typical time period over which the object must be kept, the frequency of updates, and survival across hardware or software crashes.

Supplementary Specification

- Amplify and characterize non-functional needs, quality attributes, or design constraints.
  - Examples include persistence, security features, distribution and concurrency, transaction management, and fault tolerance.
  - For each relevant category, characteristics are also captured. For example, for object persistence capture characteristics such as the number and size ranges of persistent objects, typical time period over which the object must be kept, the frequency of updates, and survival across hardware or software crashes.
  - Design constraints can also include required sequences of operations.

Glossary

- Continue adding to the Glossary.



Gather an Understanding of Architecture and Design

In the Elaboration Phase, this **gather** activity must expand and revise the understanding of the boundaries and the architectural and design constraints imposed on the candidate solutions by the broader organization's architecture, any infrastructure on which the solution would run, and any other systems with which the solution must interface. In addition, any architecture alternatives that may affect the candidate solutions are expanded to ensure that the implications are fully understood. Finally, as the architecture is explored, design alternatives will be investigated for use in **refine** to make successively more detailed design decisions in the selected solution.

It may be necessary to prototype critical external interfaces or design mechanisms that are new to the team. It may also be necessary to prototype how the architecture responds to specific quality attributes such as evolvability, security, or system throughput. Exploratory, throwaway prototypes are used to mitigate specific risks in addition to the Executable Representation that is built at the end of the iteration.

Amplify the architectural context

Architecture

- Refine the architectural context for the solution(s) in the Design Model in sufficient detail to ensure that each solution built in response to the identified stakeholder needs will operate in the larger context of the organization.
- Refine the details of the architecture constraints, boundaries, and interfaces posed by external systems with which the solution must interface. Capture this information in the Design Model and the Architecture Document.

Amplify the architectural alternatives contained in solution(s)

Alternate architectures

- Amplify architectural alternatives for further exploration and analysis during **refine** activities.
- Identify additional architectural or design techniques applicable to the solution(s).
  - In this activity, the focus is on learning techniques that might support the solution(s). This information may come from others who have solved similar problems or from the sources of the new technologies that provide design alternatives.
  - It is important to gather both the alternatives and the architectural or design implications of each alternative. This information will be used to further define the architecture for each solution.

Gather an Understanding of Marketplace and Other Sources

There are two major objectives in this phase. The first objective is to monitor the marketplace and other component sources for changes–either in the structure or direction of the source, or in new or changed components. The second objective is to continue learning as much as possible about each component (and its vendor) that is part of one or more of the candidate solutions. Using the experimentation facility, the component is further evaluated to verify vendor claims, to understand the embedded end-user business processes and architectural assumptions, and to explore functional and non-functional behaviors of the component.

Monitor relevant component sources

Look for any indications that may affect the vendor's long-term support of the components used by the solution (technology maturity, component obsolescence, component splits or mergers, vendor's going out of business, buy-outs, etc). In addition, look for new components that may drive the definition of the solution. Work with other customers to gain leverage over vendors in defining and delivering the changes, patches, and components needed for the solution.

Components

- Monitor the technologies and components used in the solution(s) for any changes or forecasts of change that are applicable to their use in the solution(s).
- Monitor the market for new technologies and components that may be relevant to the evolving solution(s).

Vendor/supplier

- Monitor the marketplace for any changes in behavior and update the relevant Market Segment Information.
  - The shape of the marketplace will change as vendors are bought out or sold and customers outsource functions or bring functionality in-house; these changes may have a bearing on the future directions components will take and are, therefore, of interest to the project.
- Review and update the vendor/supplier's market strategy, general release frequency, typical relationships with buyers, typical licensing arrangements, and long-term viability information as necessary in the Component Dossier.

Licenses

- Define the shape of any needed License Agreements for COTS components based on an understanding of the market segment, the business practices of the vendor, and the understanding of existing licenses gathered in previous iterations.

Standards

- Update any applicable standards from the broader organization, government, or appropriate standards bodies.
- Update any applicable commercial standards.
- Prioritize the standards applicable to the solution.

Evaluate applicable components

In this phase it is important to learn as much as possible about each component under consideration. For any components considered critical to the success of the solution(s), at least a demonstration version of the component must be brought to the experimentation facility (if it wasn't already there) for extensive, hands-on evaluation. Both end users and engineers should examine each of these components to understand both intended and unintended implications of using the component. Use the emerging features in the Solution Vision (for each solution) and the critical and significant use cases to fully evaluate each component.

Component capabilities

- Characterize the capabilities and limitations of candidate components in the Component Dossier with emphasis on the features in the solution(s) that are needed to meet the iteration objectives.
  - ❖ It is important to make sure that the component does what is needed in supporting both the end-user business processes and the architecture and that there are no unintended side effects from features not directly needed in the solution.
  - ❖ Testing the component to see how it responds to "bad" inputs may be important to the successful integration of this component with other components in the solution(s) or in the broader organization.
  - ❖ Watch for the impact of additional or unanticipated features on component or solution performance.

- Characterize the integration alternatives and limitations inherent in each component. Identify architectural or design issues raised by components that reflect on the critical and significant use cases.

Licenses

- Identify options for licensing COTS components.

Vendor/supplier information

- Update in the Component Dossier any changes to the component's vendor/supplier's market strategy, general release frequency, typical relationships with buyers, typical licensing arrangements, and long-term viability information.

Screen candidate Components

- Screen newly identified components against the criteria in the Component Screening Criteria and Rationale (the criteria used to remove components in earlier iterations). Capture the rationale for removing any components from further consideration in its Component Dossier.

Gather an Understanding of the Programmatics and Risks

This **gather** activity monitors the management information that defines and constrains the solution and supports tradeoffs in the **refine** activity. This information consists of cost, schedule, and risk as well as organizational policies, constraints, etc. Of particular note, because they are often forgotten, are the costs, schedule, and risks associated with implementing the business process changes that are driven by the solution(s).

The identification of risks is a pervasive task across EPIC. Risks can, and will, be identified at any point in the process and they should be documented, as described in the Risk Management Plan, as they are discovered. This gather activity provides a formal, systematic way to identify risks that may not show up in any single activity.

Update management information

Constraints

- Review and update the analysis of the gap between the skills, training, and capabilities of the team and those needed for this project.
- Monitor any changes to cost and schedule constraints.
- Monitor any changes to applicable policies.

Update procurement needs and opportunities

Contract vehicles

- Update and characterize available contracts and procurement vehicles that may be used for needed components and services for the Construction Phase and initial fielding.

Amplify implications of potential changes to end user's business process

Past technology insertion efforts

- Review and amplify the reasons for success or failure in previous efforts to change the end user's business process–especially as they relate to the changes projected in the solution(s).

Organizational change readiness assessment

- Amplify data about the culture of the target organization(s).
- Update points of leverage to induce process change.
- Learn more about the sources of resistance.

Support requirements

- Identify training requirements for all roles/levels.
- Identify help desk and technical support requirements.

Update risks

Risks

- Review and update the technical and non-technical risks in the Risk List.
  - Consider the complexity of the enterprise, the business domain to be affected, and any external constraints (e.g., cost, schedule, policy).
  - Consider risks for the solution(s) that would threaten successful development and fielding.
- Estimate potential risks for the solution that would threaten successful construction and transition.

Refine the Understanding of the Solution

The information **gathered** within each sphere of influence is synthesized with the evolving definition of the solution(s). This is accomplished by analyzing the information from the four spheres as it becomes available at lower levels of detail and by looking for mismatches between the newly **gathered** information and the existing solution(s).

The understanding of the marketplace continues to drive the definition of the solution(s). The **refine** activities evaluate how the components in the marketplace map to what the users want as well as how the components relate to each other in the context of the user needs at successively lower levels of detail. As more is learned about the capabilities and limitations of available components, some components may be dropped from further consideration; some Stakeholder Requests and/or end-user business processes may be modified, or some candidate solutions may be dropped entirely.

Over the course of several iterations, one solution must be built to the point that the Solution Vision is clearly articulated, the design is stable, the implications for the end user's business process are identified, and the stakeholders have agreed to implement any necessary changes. Early in the Elaboration Phase, candidate solutions will be amplified sufficiently to allow the one solution to be selected. In later iterations, the selected solution will be amplified sufficiently for implementation in the Construction Phase. The iteration tasks described below apply to both early and late iterations; they will use *solution(s)* to represent both the candidate solutions (early iterations) and the selected solution (late iterations).

In the Elaboration Phase, there may be one or more exploratory, throwaway prototypes during **refine** to mitigate specific risks such as design and stakeholder needs tradeoff, component or component feasibility analyses, or demonstrations of key scenarios to certain stakeholders. In this phase is it important to build a prototype of the user interfaces and conduct end-user feedback sessions if this was not done in conjunction with analysis of a critical Use Case in the Inception Phase. In general, any high-risk architectural and design issues associated with the solution(s) should be demonstrated to show feasibility and consistency with the external interfaces and infrastructure.

Refine is not composed of sequential tasks. Rather, the tasks listed below represent work that is occurring concurrently. All tasks are both dependent on and critical to the other tasks with continuous feedback among them. The four spheres of **gathered** knowledge, and the models that represent that knowledge, are addressed simultaneously to amplify the solution(s). The emphasis in **refine** is to identify and resolve mismatches across the information from the disparate spheres of influence. Incomplete information is resolved by requesting that additional information be **gathered**. Conflicts in information are determined through analysis and resolved through negotiation.

Identify and resolve mismatches from the synthesis of new information

The following steps analyze the new information from the various **gathers** in the context of the existing solution(s). The focus in this task is on understanding and resolving, if possible, any mismatches between the newly gathered information and the solution(s) as it is currently defined. It is important to understand how important each mismatch is to the solution(s) and understand the possible ways the mismatch can be resolved. Mismatches can be resolved through negotiating with stakeholders to modify end-user business processes and stated stakeholder needs, gathering more information about the capabilities of the components and their ability to be tailored to accommodate the mismatch, changing the way the architecture uses the components, or creating a custom component to provide the necessary capability. If a mismatch cannot be sufficiently resolved, the solution(s) may be removed from further consideration.

The steps below describe the basic steps that must be completed, but the order will be subject to the needs of a particular iteration; they will seldom be implemented in the order shown. In most cases, cycling between the steps will be required as resolution of mismatches in one sphere introduces changes to the baseline and, therefore, potential new mismatches in another sphere.

End-user business processes

- Incorporate applicable new information about end-user business processes in the solution(s). Identify and characterize the nature of any mismatch.
- Incorporate applicable new information about the broader organization business processes in the solution(s). Identify and characterize the nature of any mismatch.
- Resolve identified mismatches, where possible, through negotiations with the affected stakeholders. Record the results of the negotiation in the Target Business Use-case Model or the Target Business Object Model as appropriate.

Use Cases

- Incorporate applicable new information about critical and significant Use Cases in the solution(s).
- Review and identify mismatches where the solution(s) falls short in meeting the needs of the critical and significant Use Cases. Determine the impact of not meeting this need.
  - The Use Cases include both functional behaviors and the quality attributes (like performance) that are specific to a use case.
  - It may be useful to understand how other customers with similar needs address these shortfalls.

- Review and identify mismatches where the solution(s) operates differently from the behavior required in the critical Use Cases. Determine the impact of changing end-user business processes to match the solution.

- Review and identify mismatches where the solution(s) provides functionality not addressed in the Use-case Model. Determine if the additional features adversely affect end-user business processes or, perhaps, offer an opportunity to optimize end-user business processes.

- Resolve identified mismatches, where possible, through negotiations with the affected stakeholders. Record the results of the negotiation in the Target Business Use-case Model, the Target Business Object Model, the Use-case Model, and the Supplementary Specifications as appropriate.

  - Consider techniques such as Win-Win Requirements Negotiation [13, 14] or any of the techniques described in *Getting to Yes* [15].

- Update the definition of any custom components necessary to resolve mismatches that cannot be negotiated.

Non-functional

- Incorporate applicable new information about the use-case mechanisms and Supplementary Specification in the solution(s).

- Review and identify mismatches where the solution(s) falls short in meeting the needs identified in the use-case mechanisms and the Supplementary Specification. Determine the impact of not meeting this need.

  - The Supplementary Specification includes the quality attributes of the solution that are not specific to a use case.

- Review and identify mismatches where the solution(s) operates differently from the behavior required in the use-case mechanisms or the Supplementary Specification. Determine the impact of changing end-user business processes to match the solution.

- Review and identify mismatches where the solution(s) provides quality attributes not addressed in the use-case mechanisms or the Supplementary Specification. Determine if the additional attributes adversely affect end-user business processes or, perhaps, offer an opportunity to optimize end-user business processes.

  - An example might be a component that makes security provisions that had not been considered in the original stakeholder needs.

- Resolve mismatches, where possible, through negotiations with the affected stakeholders. Record the results of the negotiation in the Use-case Model and Supplementary Specifications.

  - It may be possible to narrow the application of a particular quality attribute (as when the attribute must be present only under particular circumstances or in particular parts of the solution).

- Update the definition of any custom components necessary to resolve mismatches that cannot be negotiated.

Architecture

- Incorporate applicable new information about the architecture in the solution(s).

  - The architectural information includes external interfaces, existing infrastructure, design constraints, and architectural and design alternatives.

- Review and identify mismatches where components within the solution(s) do not have needed interfaces or behaviors to link effectively into the architecture, with each other, with existing infrastructure, or to the broader organization's architecture. Determine the impact of not meeting these interfaces or behaviors.

- Review and identify mismatches where components overlap. Determine the ease with which certain functionality can be bypassed.

- Review and identify mismatches where the components provide interfaces not addressed in the architecture. Determine if the additional interfaces adversely affect the architecture or, perhaps, offer an opportunity to optimize the architecture.

- Resolve mismatches, where possible, through negotiations with the affected stakeholders. Record the results of the negotiation in the Design Model.

  - In this case, the affected stakeholders are likely to be vendors and other suppliers, architects, senior designers, or infrastructure owners.

- Update the definition of any custom components, wrappers, or integration mechanisms ("glue") necessary to resolve mismatches that cannot be negotiated.

  - You need to know if "glue" will be needed (and if so, a rough order of magnitude of how much) and what kind of component integration strategy might link the components.

Components

- Incorporate applicable new information about the components in the solution(s). (The following steps will identify and characterize the mismatch.)

- Review the coverage of components against the amplified Use Cases. Amplify the understanding of where the components overlap, where the sum of the components is deficient, and where the components provide functionality that is not currently requested.

  - Deficiencies in component coverage will have to be addressed either by changing the scope (which may entail a reconsideration of the Life-cycle Objectives anchor point) or by building custom components.

- Review how components support the mechanisms and services needed by the critical, significant Use Cases and Supplementary Specification.

- Review where new or changed components appear to fit in the architecture and where there are potential mismatches with the existing infrastructure and external interfaces.
  - ❖ Look at existing networks (e.g., protocols), databases, database designs, firewalls, servers, and naming standards.

Cost and schedule

- Incorporate applicable new information about the costs and schedule of the solution(s). Identify and characterize any mismatches.
- Resolve any mismatches among cost, schedule, and the capabilities in the candidate solution through negotiation with the affected stakeholders. Record the results of the negotiation in the Project Plan, the Use-case Model, the Supplementary Specifications, and/or the Design Model as appropriate
  - ❖ The resolution to mismatches with cost or schedule includes relaxing the cost or schedule constraint or removing capability from the solution.

Screen candidate solutions

- Any candidate solutions with mismatches that could not be negotiated should be considered for removal from further consideration. Capture the rationale for removing any candidate solutions in the Business Case.

Amplify the solution(s)

The following steps mature the definition of the solution(s). In the process of amplifying the solution, decisions will be made resulting in selecting specific components, and perhaps, redefining the architecture, making further changes to stakeholder needs, and modifying end-user business processes. The solution(s) is defined by *what* the solution(s) will do and *how* the solution(s) will be implemented. The *what* and the *how* are defined concurrently with tradeoffs between them. Defining the Solution Vision—the *what*—and the architecture and the management of changes to the end user's business process—the *how*—such that they can be validated and baselined, is the primary objective of this set of activities. The steps that follow are performed concurrently with feedback necessary between the steps.

The Solution Vision and the Solution Requirements Specification that evolves from it describe *what* the solution will do. The Solution Vision is the high-level customer view of the solution describing those features that must be present. The Solution Requirements Specification consists of Use Cases, the Supplemental Specification, and any "must have" requirements necessary to specify the solution. The term *requirement* is reserved for those capabilities that must exist in order for the solution to meet the mission need–the "must haves" or "hard" requirements. The requirements describe the functional and non-functional capabilities that must be in the

fielded solution (i.e., the minimum success criteria for the solution.) "Important-to-have" or "nice-to-have" capabilities that enhance the ability of users to accomplish the mission need but are not mandatory for meeting the mission need are not carried as requirements, but are maintained in the Stakeholder Requests. There should be a limited number of requirements to maintain a sufficient amount of trade space to leverage component capabilities.

The architecture as reflected in the Design Model and the Architecture Document describes *how* the Solution Vision will be implemented. The architecture must accommodate the critical use cases, candidate components, the interfaces to the broader organization's architecture, applicable architecture standards, and any identified design constraints. The architecture must also take into consideration alternative designs and trades between requirements, interactions of components, and impacts to end-user business processes. The architecture for a given solution will be driven by the architectural assumptions of each of the included components.

Over the life of the architecture, components will have to be updated continuously, and, less frequently, technology may have to be replaced. The architecture for the solution(s) must be flexible enough to accommodate the current components and the projected growth path for those components to optimize the solution's ability to evolve efficiently as components and technologies change. Evolvable architectures exhibit many of the following characteristics:

- specialized layering: logical groupings of components where the upper layers are designed with more specialized domain knowledge whereas lower layers are less domain specific
- highly modular: self-contained, understandable components that are optimally scoped and sized for non-functional requirements and attributes such as maintainability
- well-defined interfaces: focus on the essential characteristics of the component and hide the details of a specific component
- standard interfaces: industry standard interfaces that promote component exchangeability
- common mechanisms: mechanisms used consistently wherever possible, such as inter-process communication, user-interface interaction, and error handling

Management of any necessary end-user business processes is a key part of the defined solution(s). Identification of the changes necessary and a determination of how various parts of the organization will implement those changes must coincide with the definition of the architecture.

Solution Vision

- Amplify the Solution Vision of what is/is not intended in the solution(s).
  - The Solution Vision should reflect a solid understanding of the critical and significant Use Cases and the components included in the solution(s) that drive the architectural and planning decisions.
- Amplify the features associated with the solution(s) (with growth vectors, quality attributes, and priorities).
  - The features will depend on the increasingly detailed understanding of the components that make up the solution and the capabilities they offer as well as the increasingly detailed understanding of the stakeholder's needs.
- Gain agreement with the stakeholders on the Solution Vision.

Requirements

- Analyze the features and components in the context of the critical and significant Use Cases and the Supplementary Specification to identify testable functional and non-functional requirements associated with the solution(s).
  - Requirements associated with a particular Use Case are captured as part of the Use Case.
  - Requirements not suited to Use Cases are captured in text, tables, or other diagrams as attachments to use-case constructs and/or in the Supplementary Specification as part of the Solution Requirements Specification.

Architecture

- Amplify (or create) a Design Model, which captures the high-level partitions (or analysis "packages") and the relationships among them, to establish the basic structure of the architecture.
  - Ensure that the analysis packages and their relationships continue to hold true as the architecture is defined at increasing levels of detail. Define new relationships where needed.
  - When creating the initial Design Model, use previously identified components and use-case mechanisms and services as a starting point.
  - The architecture must address the critical non-functional requirements as well as the functional requirements.

- ❖ Look across the Use Cases and components to identify and understand shared infrastructure needs, types of interactions required of resources, and other parts of the solution.

- ❖ To better support the ability of the solution to evolve, distinguish between packages that describe attributes that are general to a class of problems from those that are truly specific to this particular solution.

- ❑ Elaborate the Design Model with the architecturally significant aspects of the components, the external interfaces, and any existing infrastructure.

  - ❖ The components will drive the structure and nature of the architecture.

- ❑ Update and design any new custom components needed to satisfy requirements not covered by components.

- ❑ Develop and update the detailed characterization of the interactions of the components and a component integration approach that identifies how components, custom components, legacy, and the broader organization's architecture will be linked.

- ❑ Design any custom code necessary for integrating the solution(s) (tailoring, glue, wrappers).

- ❑ Update the Design Model to reflect multiple architectural views. The architecture should reflect all needed views of the solution.

  - ❖ The logical view captures the functional requirements of the system, that is, what the system does for the end users. The logical view identifies the major design packages and subsystems, such as the package that creates flight plans for an air traffic system.

  - ❖ The process view captures the concurrent elements of the system at runtime, such as tasks, threads, startup, shutdown, throughput, and fault tolerance.

Waivers

- ❑ Resolve any waivers necessary for "compliance" with the broader organization's architecture or other applicable policies.

Architecture Document

- Update the salient features of the Use Cases and Design Model in the Architecture Document.
  - The Architecture Document contains various high-level architectural views of the system, key decisions, and lessons learned.
- Capture key decision made in defining the architecture (e.g., mechanisms tried and abandoned, components discarded).
- Record approved and denied waivers in the Architecture Document.

Changes to the end user's business process

- Amplify the end-user business processes that will be supported by the solution(s). Capture, using the Target Business Use Case and Object Models, the processes along with the workers, their responsibilities, and the operations they will perform.
- Assess potential changes to end-user business processes against the use cases in the context of the evolving architecture.
- Ensure affected stakeholders understand the scope of the changes to the end user's business process and are committed to the necessary change implementation.

Business Process Change Management Plan

- Review and update the delta (width and depth) from current to target end-user business processes represented in this solution.
- Review and update the migration strategy and the high-level plan that migrates the affected organizations from the current to the target end-user business processes.

Cost and Schedule

- Estimate total ownership costs and schedule for the solution(s) across its life.
  - Capture any attributes of the test approach for the solution(s) that will drive cost or schedule.
  - Capture any attributes of the fielding approach for the solution(s) that will drive cost or schedule.
  - Capture any attributes of changes to the organization/end-user business process for the solution(s) that will drive cost or schedule.

Risks

- Characterize and analyze the identified risks for this candidate solution.

Business Case

- Update the general functionality, performance, quality, fielding approach, changes to the end user's business process, cost/benefit, schedule, and risks of the solution(s) over its anticipated life.

Screen solutions

- If necessary, screen solutions to a reasonable number for further consideration.
  - It is important that the Business Case shows that all possible alternatives were explored. The objective in this phase, however, is to select one solution to be built, fielded, and supported. As candidates are differentiated in terms of satisfaction of critical and significant Use Cases, solution(s) can be screened to allow for focused attention on the most likely solution(s).
- Record the rationale for eliminating any solution(s) in the Business Case.

Assemble an Executable Representation

In this phase, evolutionary prototypes are built and amplified as more is learned in successive iterations. These evolutionary prototypes provide an opportunity for both the end users and the engineers to evaluate the solution(s) based on "hands-on" experience. An evolutionary prototype is the vehicle for integrating and assessing selected architectural components against primary scenarios. It supports the analysis of interface and behavior details of relevant component(s) and demonstrates component integration mechanisms.

By the end of the Elaboration Phase, the evolutionary prototype will demonstrate the adequacy of the Solution Vision, validate that the selected solution satisfies the Solution Vision with manageable risk, and serve as the baseline for the Construction Phase. The evolutionary prototype also provides a vehicle to prototype end-user business processes so that affected stakeholders have sufficient insight to understand and agree to any necessary changes to the end user's business process. In addition, affected stakeholders can use the evolutionary prototype to determine that the solution is adequate to meet the mission needs.

Build and test an architectural prototype

In this phase, it is more important that the Executable Representation demonstrate the iteration's objectives than that it be constructed with full engineering rigor. However, the evolutionary prototype is not a throwaway; it will serve as the baseline for the engineering activities in the Construction Phase. Deliberate trades must be made between building the prototype quickly and providing enough rigor to support this long-term evolution to an operational solution.

Implementation Model	❑ Define the organization of the Executable Representation in terms of implementation subsystems and/or components organized in layers.
Test Set Artifacts	❑ Develop test scripts and other Test Set Artifacts needed to evaluate the Executable Representation.
	❑ Build/update test cases, drivers, stubs, etc.
Implement and test components	❑ Write source code, adapt existing components, compile, link, test, and execute.
	❑ Submit rework feedback on the design if defects are discovered.
	❑ Tailor and test the components.
	❑ Develop and test the integration code or data (wrappers, glue, data sets, etc.) needed to incorporate pre-existing and custom components.
Integrate and test	❑ Integrate new and changed components into a new version.
	❑ Perform integration tests.
Deployment Artifacts	❑ Draft the End-user Support Materials to reflect any updates of the Use-case Model in light of the negotiated changes.

Prototype the needed changes to the end user's business process

As the architectural prototype is being assembled, the end users must prototype the changes to the end user's business process associated with the solution(s).

Change management

- Prototype needed change(s) to the end user's business process.
- Prototype the appropriate elements of the Business Process Change Management Plan.

Assess the Iteration

As the iteration is completed, it is important to determine if the objectives planned for this iteration were achieved (unresolved issues will be assigned to future iterations). In addition, a review of any unplanned questions, risks, or issues that arose during the iteration must be conducted so that they can be captured in the appropriate planning artifacts.

In addition, based on all of the activities in the iteration, the new harmonized information within the solution(s) is summarized in the Business Case. In the Elaboration Phase, the Business Case captures the rationale for determining which solution is of sufficient interest for implementation in the Construction Phase.

Assess the architectural prototype for the solution(s)

The Executable Representation provides an opportunity for both the end users and the engineers to evaluate the evolving solution in the context of the objectives for the iteration. End users must verify that the end-user business processes represented are acceptable. Engineers must verify that the technical design can be implemented. Together they verify that the iteration's objectives have been met, that the evolving solution meets real needs, that the end-user business processes proscribed by the solution are acceptable, and that the solution can be made to operate in an acceptable manner.

- Validate the solution.
  - Show that the solution is what the stakeholders need or want.
- Verify that the solution is implemented correctly.

Update the information about the solution

Screen candidate solutions

- Define and update the criteria that will be used to select a single solution.
- When there is enough information, screen candidate solutions to select a single solution to be built in the Construction Phase.
  - ❖ Solutions may be screened in any iteration as information is discovered that shows that one or more solutions is not suitable. Once a single solution has been selected, iterations will skip this activity.

Business Case

- Amplify the Business Case to capture the significant decisions made in the iteration. In particular, the rationale for selection of a single solution should be captured.

Determine lessons learned from iteration

Iteration Assessment

- Determine if objectives planned for this iteration were achieved (selected risks mitigated).
  - ❖ Unmet objectives will be assigned to future iterations.
- Identify any unplanned questions, risks, or issues that arose during the iteration and assign to future iterations.

Risk List

- Update the Risk List based on the Iteration Assessment.
- Identify mitigation approaches for the priority risks.

Project process improvement

- Review project metrics and make recommendations for process improvement.

Assess the phase, if the iteration completes the phase

Assessment group

- Appoint an assessment group with representatives for all affected stakeholders, including end users.

Phase exit criteria

- The assessment group determines whether the phase objectives and exit criteria have been met and decides whether the project should go ahead. This constitutes the LCA anchor point.

  - The phase exit criteria should have been documented in the Project Plan. Much of the necessary information that shows whether or not those criteria are met should have been captured in the Business Case.

- The assessment results are captured in the phase Review Record.

Supporting Activities

The activities associated with project monitoring and control, technical process activities, and supporting process activities that were included in the Development Plan have not been described in the activities laid out in this document. They are, however, critical to the success of the project.

Monitor project status

Project progress

- Monitor the progress of the project relative to the Project Plan from the viewpoints of the various stakeholders (including budget and schedule).
  - ❖ This includes the monitoring metrics that measure the progress of the solution.
  - ❖ Capture and assess any measurements associated with the project's measurement goals. Verify that the right goals are measured.

Exceptions and problems

- Seek out exceptions and problems that must be resolved for project success.

Prepare experimentation facility

Prepare the experimentation facility in accordance with the Infrastructure Plan to support the identified tasks for this phase and the Construction Phase.

Update and create contracting vehicles as necessary

Appropriate procurement vehicles must be in place to support the ongoing activities in the Elaboration Phase and the projected activities in the Construction and Transition Phases.

Contracts and procurement vehicles

- Update and create all contracts and procurement vehicles for needed components and services.
  - ❖ Prepare the Justification and Approval for selected COTS components or appropriate documentation for negotiating needed contract vehicles.

License Agreements

- Negotiate licenses.
  - ❖ Whereas demonstration licenses are common in the Elaboration Phase, development licenses will be needed in the Construction Phase and runtime licenses will be needed for the end users in the Transition Phase.

Phase Artifacts

The major artifacts from the Elaboration Phase capture, demonstrate, and validate that

- the agreed-upon solution harmonizes the stakeholder needs and end-user business processes, the selected components, the architecture and design, and cost and schedule
- the agreed-upon solution has been defined with sufficient fidelity for the Construction Phase
- the risks have been identified and mitigated sufficiently

TO CHARACTERIZE END-USER BUSINESS PROCESSES AND STAKEHOLDER NEEDS

Current Business Use-case Model
Current Business Object Model
Target Business Use-case Model
Target Business Object Model
Glossary
Stakeholder Requests
Solution Requirements Specification
Use-case Model
Use Cases
Supplementary Specification

TO CHARACTERIZE THE MARKETPLACE

Market Segment Information
Component Dossier (for each examined component)
Component Screening Criteria and Rationale

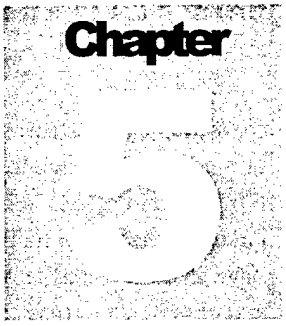
TO CHARACTERIZE THE ARCHITECTURE AND DESIGN

Solution Vision
Design Model
Architecture Document
Executable Representation(s)
▪ Implementation Model (for each Executable Representation)
Test Set Artifacts
Deployment Artifacts (includes End-user Support Materials (optional in this phase))

TO CHARACTERIZE PROGRAMMATICS AND RISK

Development Plan (All artifacts in the Development Plan are reviewed and updated in each iteration.) The artifacts listed below are of particular interest in this Phase.
▪ Project Plan
▪ Iteration Plans
▪ Risk Management Plan
▪ Process Improvement Plan
▪ Development Case
▪ Infrastructure Plan
▪ Vendor/Supplier Relationship Plan
Deployment Plan
Business Case
Business Process Change Management Plan

Risk List
License Agreements
Iteration Assessments
Review Record



Construction Phase

The goal of the Construction Phase is to achieve a production-quality release ready for its user community. The selected solution is prepared for fielding.

Phase Description

The focus of the Construction Phase is on preparation of a production-quality release of the selected solution approved at the LCA anchor point that is suitable for fielding. Any custom components needed are developed. Production rigor is applied to component tailoring, integration code or data (including wrappers, glue, data sets, etc.) needed to incorporate pre-existing and custom components, and system testing. Additionally, the Construction Phase includes preparation of necessary support materials, such as installation instructions, version descriptions, user and operator manuals, and other user and installation site support required.

The Construction Phase continues the preparation of the end-user business environment of the target organizations to facilitate the initial fielding of the solution. This preparation includes development of required policies and procedures, restructuring of the organization as necessary, implementation of the changes to the end user's business process for the initial rollout groups, and the establishment of incentives, user groups, and other mechanisms to encourage adoption of the solution.

While every effort is made during the Elaboration Phase to stabilize the solution and to address risks, inevitably some unanticipated changes may occur in requirements, components, and the architecture and design during the Construction Phase. In particular, because of the volatile nature of the marketplace, new versions of the selected components will require detailed investigation as suppliers add, change or remove functionality. Continued monitoring of the marketplace and evaluation of new and changed components is required to anticipate changes and determine an appropriate component upgrade approach.

The experimentation facility created to support the Elaboration Phase will continue to be needed to evaluate new and changed components. The risk to the cost and schedule of the

Construction Phase presented by any change will have to be balanced against the risk of not upgrading and delivering obsolete components. For minor changes, the construction can be temporarily delayed while adjustments are identified, validated, and implemented. For major changes, decisions made at the LCA anchor point (or even the LCO anchor point) may have to be revisited or the changes deferred to the next generation solution.

With an emphasis on assembling a production-quality release of the solution, the principal artifacts of this phase are

- the Executable Representation in the form of a production-quality release of the solution suitable for fielding to a limited set of end users in an initial rollout (or beta)
- the Deployment Plan that captures how and when the solution is to be made available to the user community
- the Test Artifacts that plan and capture information associated with tests performed to assess the quality of the solution.

The Construction Phase ends with the Initial Operational Capability (IOC) anchor point. The IOC anchor point allows stakeholders to verify that a production-quality release of the solution is ready for fielding to at least a subset of the operational users as an initial fielding or beta test.

Phase Objectives

Implement the selected solution with production quality to operate within the broader organization's architecture.

Minimize engineering costs by optimizing resources and avoiding unnecessary scrap and rework.

Achieve adequate quality as rapidly as practical.

Achieve useful versions (alpha, beta, and other test releases) as rapidly as practical.

Maintain current marketplace information.

Balance engineering stability and potential component obsolescence with marketplace volatility.

Prepare initial fielding end users for beta test (e.g., implement, or prepare to implement, changes to the end users' business process; complete training; manage resistance).

Put contract vehicles in place for the Transition Phase.

Phase Task Overview

Phase Planning Activities
- Update the Development Plan for the project

Iteration Activities

Plan the Iteration
- Build a detailed plan for the iteration
- Update the Development Plan for the project

Gather ...

... an Understanding of Stakeholder Needs and End-User Business Processes
- Update and expand the business model as necessary
- Capture the significant behaviors of the solution

... an Understanding of Architecture and Design
- Review, and update as needed, the architectural context

... an Understanding of Marketplace and Other Sources
- Monitor relevant market segments
- Characterize component changes

... an Understanding of the Programmatics and Risks
- Update management information
- Update procurement needs and opportunities
- Monitor implications of changes to the end user's business process
- Update risks

Refine the Understanding of the Solution
- Identify and resolve mismatches from the synthesis of new information
- Update the solution if needed

Assemble an Executable Representation
- Build and test the solution
- Implement the needed changes to the end user's business process
- Make any needed existing infrastructure and external interfaces changes

Assess the Iteration
- Assess the Solution
- Update the information about the solution
- Determine lessons learned from iteration
- Assess the phase, if the iteration completes the phase

Supporting Activities
- Monitor project status
- Maintain the experimentation facility
- Update and create procurement vehicles as necessary

Phase Exit Criteria

Status	Exit Criteria
	Affected stakeholders agree the solution baseline, as demonstrated in the Executable Representation, is mature enough to be fielded in the user community (the release is stable). ▪ Existing defects are not obstacles to achieving the purpose of the release. ▪ Pending changes are not obstacles to achieving the purpose of the release.
	Affected stakeholders approve the defined and documented procedures for transition of the solution to the user community (and have implemented the procedures for users affected by initial fielding).
	Relationships with vendors are adequately managed.
	Information on relevant components and the marketplace is current and recorded.
	Any differences between actual resource expenditures versus planned expenditures for this phase are understood, and corrective actions are included in the Project Plan.
	The experimentation facility is sufficient to support continued monitoring of the components and relevant market segments.
	The Development Plan has been updated. (The plan for the Transition Phase is sufficiently detailed and accurate and is backed up with a credible basis for all estimates.)
	The Deployment Plan is sufficient to support beginning the Transition phase.
	The contract vehicles are in place for initial fielding and in progress for full fielding.

Phase Planning Activities

Update the Development Plan for the project

The Development Plan was updated in the Elaboration Phase to include detailed plans for the Construction Phase. The Development Plan references the entire set of major planning artifacts that describe the project and how it will be executed. This plan will continue to be updated in each iteration as more is learned about the solution. In general, iterations in this phase focus on creating a production-quality solution, developing Deployment Artifacts, and implementing any end-user business processes necessary for fielding to operational users. At the end of the Construction Phase, a detailed plan for the Transition Phase is prepared and the Development Plan is updated accordingly.

Development Plan	❑ Review the project organization. Verify that the level of staff resources and skills available are appropriate.
	❑ Update the Project Plan.
	❑ Review the plans for project monitoring and control and update as needed.
	❑ Update the Risk Management Plan.
	❑ Update the Development Case if needed.
	❑ Review the remaining plans for technical process planning and update as required. In particular, update the planning for the experimentation facility in the Infrastructure Plan.
	❑ Review the Vendor/Supplier Relationship Plan; ensure that appropriate information channels appropriate to the importance of each major vendor/supplier are maintained.
	❑ Review the plans for the other supporting processes and update as needed.
Test planning	❑ Update the Solution Acceptance Plan if needed. Validate any changes with stakeholders.
	❑ Update test objectives and plan in the Test Set Artifacts.
Plan for fielding	❑ Amplify all activities performed in fielding the solution to the customer. Activities include planning, beta testing, preparing items to be delivered, data migration, packaging, shipping, installation, training, and support.

- ❖ Review and update the responsibilities of both the customer and the development team in preparing for fielding. (Of particular relevance in this section is the description of the customer's involvement in acceptance tests and the process for handling any discrepancies.)

- ❖ Review and update the schedule and milestones to conduct the fielding activities. Fielding milestones need to conform to the project milestones.

- ❑ List the resources (and their sources) required to carry out the planned fielding activities.

  - ❖ Review and update the facilities required to test and field the solution. (Facilities may include special buildings or rooms with raised flooring, power requirements, and special features to support privacy and security requirements.)

  - ❖ Review and update the hardware required to run and support the solution. Specify model, versions, and configurations. Provide information about manufacturer support and licensing.

  - ❖ Review and update the list of the software and documentation provided as part of the deliverable solution.

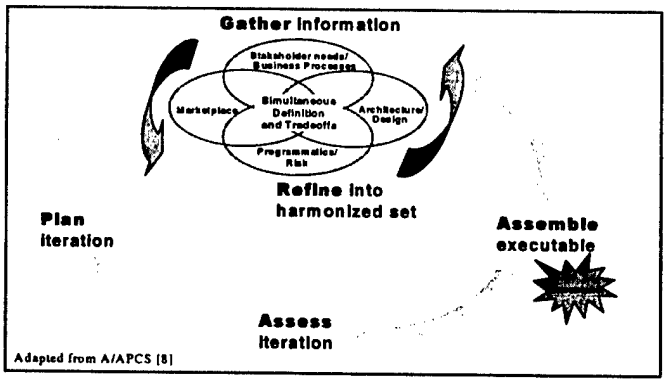
- ❑ Review and update the plan and inputs for training the end users such that they can use and adapt the solution as required

Planning for changes to the end user's business process

- ❑ Update and validate with end users a detailed Business Process Change Management Plan for needed changes to organizational structure and/or end-user business processes to match the Deployment Plan.

Iteration Activities

ITERATION ACTIVITIES KEY
Plan
Gather
Refine
Assemble
Assess



Plan the Iteration

Plan the iteration in detail at the beginning of each iteration. Any lessons learned and new risks discovered from previous iterations are incorporated in the Risk List, which is a key consideration for defining the iteration objectives and activities. The information (and the level of detail of that information) to be **gathered** about each of the spheres of influence is determined by the iteration objectives. The iteration objectives also determine what analysis must be done to **refine** the information from each of the spheres. Of more importance in this phase, the iteration objectives determine the content of the Executable Representations to be **assembled**.

Build a detailed plan for the iteration

The detailed plan for the iteration is based on the current Risk List and the most critical functionality. Risk is a key discriminator in deriving objectives for the iteration. The highest priority risks are mitigated as early as possible. However, addressing risk is balanced with ensuring that the critical functions and services the solution must provide are addressed early.

Iteration Plan

- Refine the scope of the iteration and the goals and objectives that were planned in the Project Plan for this iteration to reflect any changes since the plan was last updated.
- Define objectives for the success of the iteration.
  - These objectives will provide focus for all activities in the iteration and will be used at the end of the iteration to decide whether the iteration was a success.

- Define the requirements for the Executable Representation needed to demonstrate the iteration objectives.
  - This *evolutionary* prototype is created in addition to one or more *exploratory*, throwaway prototypes to mitigate specific risks such as design and requirements tradeoffs, component feasibility analyses, or demonstrations of key scenarios to certain stakeholders conducted as part of **gather** or **refine**.
  - Depending on the specific objectives of the iteration, more than one prototype may be needed.
- Identify the tasks that will be required to achieve the iteration objectives and the specific artifacts that must be developed or updated.
- Complete a detailed work breakdown structure to show how the work that must be done within the iteration is allocated and what resources are necessary to do the work.
- Determine milestones (events and dates) that are important to the iteration.

Update the Development Plan for the project

Development Plan

- Revise and update the Development Plan.
  - The plan should be updated to reflect changes to the technical or programmatic baseline, to reflect changes in personnel availability or skills, to reflect the changes necessary to accommodate a particular set of components, or simply to reflect a new approach to meeting the identified needs.

Gather Information

There continues to be a set of **gather** activities that collect the detailed implementation information from each of the four spheres needed to meet iteration objectives. In addition, the **gather** activities continue to monitor the four spheres for any changes that affect the solution being built. The information from one sphere continues to depend on and drive the information needed from another sphere. Therefore, once a change has been identified in one sphere, it is likely that the **refine** tasks for the iteration will require additional **gather** activities to collect complementary information from the other spheres of influence to meet the iteration objectives. The **gather** and **refine** activities will continue to cycle until the information becomes sufficiently detailed to meet iteration objectives.

Gather an Understanding of Stakeholder Needs and End-User Business Processes

This phase will continue to define the detailed needs, boundaries, and constraints of the solution. This definition will change due to new or changed information from the stakeholders or may be driven by additional insight into the components. This information will still expand the Use-case Model and the documented stakeholder needs that support the Use Cases. In addition, any changes in the business model may require that associated Use Cases be reviewed and validated.

Update and expand the business model as necessary

The business model is described in the Target Business Use-case Model and the Target Business Object Model. In the Construction Phase, review and update end-user business processes that affect or are affected by the solution.

Mission	❑ Review and update the understanding of the mission and strategic direction of the end-user organization and the forecast of potential changes and forces of change that will affect the end-user organization and its mission over time.
	❑ Update any business goals of the end-user organization that relate to, or are affected by, the solution.
Stakeholders	❑ Review any changes in stakeholders identified with a vested interest in the outcome of the solution.
Current fielding capabilities	❑ Review, and update as necessary, the characterization of the current operational environment for each of the relevant stakeholders who will use some aspect of the solution.

- Verify that current fielding capabilities are understood as well as any methods and resources currently in place to implement system upgrades or major enhancements.

Business measures

- Review, and update as necessary, the measurable business improvement goals and the metrics that will be used to evaluate the solution.

Business modeling

- Review, and update as necessary, the Current Business Use-case Model and Current Business Object Model to support transition to the solution.
- Review, and update as necessary, the Target Business Use-case Model and Target Business Object Model to reflect the implications of the solution.
- Review, and update as necessary, the Glossary.

Capture the behaviors of the solution

The purpose of this task is to review and update any new information from the stakeholders regarding what their needs really are. The characterizations of needs for the solution should be very stable in this phase. Changes may be necessary if there are significant component changes during the Phase. Maintain agreement with the stakeholders and set realistic expectations on what will be delivered.

The Supplementary Specifications capture the non-functional behaviors and quality attributes of the solution that are not readily captured in the Use Cases (for example, legal and regulatory requirements and application standards; quality attributes of the solution, including usability, reliability, performance and supportability; and other requirements such as operating systems and environments, compatibility requirements, and design constraints).

Stakeholder Requests

- Review and update Stakeholder Requests that relate to the solution.

Use-case Model

- Review and update the complete set of Use Cases; validate that the right set has been captured; revisit the prioritization of the Use Cases.

Use Cases

- Review and update existing Use Cases and complete any remaining undefined Use Cases that affect or are affected by the solution.
  - ❖ Adapt the architecturally non-significant Use Cases to work in the defined solution.

Use-case mechanisms

- Review and update the mechanisms and services that are needed by the Use Cases.

Supplementary Specification

- Review and update the non-functional needs, quality attributes, or design constraints in the Supplemental Specification.

Gather an Understanding of Architecture and Design

In the Construction Phase, this **gather** activity monitors the broader organization's architecture, infrastructure, and interfaces to external systems for changes that may affect the solution.

Review, and update as needed, the architectural context

Architecture

- Review and update the details about the architecture constraints, boundaries, and interfaces (including interfaces to external systems and existing infrastructure).

Gather an Understanding of Marketplace and Other Sources

In this phase it is important to continue to monitor the marketplace for changes–either in the structure or direction of the market, or in new or changed components. Any new or changed components identified will be examined in the experimentation facility to determine how they might impact the release that is being built. The focus in this Phase is on implementing the solution so any decision to change components will have to be carefully balanced against the need to produce a release for fielding.

Monitor relevant market segments

Look for any indications that may affect the vendor's long-term support of the components used by the solution (technology maturity, component obsolescence, component splits or mergers, vendor's going out of business, buy-outs, etc.). In addition, look for new components that may drive the definition of future solutions and work with other customers to gain leverage over vendors in defining and delivering the changes, patches, and components needed.

Market Segment Information

- Monitor the technologies and components used in the solution for any changes or forecasts of change.
- Monitor the market for new components that may be relevant to the solution.

- Monitor the buying habits of the other buyers, the delivery of component releases, the behavior of the vendors/suppliers, and size and distribution of each relevant market segment to track and forecast trends.
- Update the relevant Market Segment Information.

Characterize component changes

In this phase there will likely be changes to the selected components. In addition, new components may have been introduced in the marketplace. A copy of each component, release, or patch should be obtained and loaded in the experimentation facility for detailed examination. Both end users and engineers should examine each changed component to understand how any changes might affect the solution.

Component capabilities

- Characterize the capabilities and limitations of new and changed components in the Component Dossier.
  - ❖ It is important to make sure that the component still does what is needed in supporting both the end user's business processes and the architecture and that there are no unintended side effects from any changes.
- Identify any impacts on the architectural or end-user business process that result from any component change.

Licenses

- Monitor and update licensing options information for COTS components.

Vendor/supplier drivers and health

- Continue to monitor the vendor/supplier's market strategy, general release frequency, typical relationships with buyers, typical licensing arrangements, and long-term viability. Update the Component Dossier as necessary.

Screen candidate components

- Screen newly identified components against the Component Screening Criteria and Rationale (the criteria used to screen components in earlier iterations) capturing the rationale for removing any components from further consideration in the Component Dossier.

Gather an Understanding of the Programmatics and Risks

It remains important to monitor changes in information developed and maintained external to the project (i.e., policy directives, budget, and sometimes schedule and/or priorities) as well as to capture the programmatic information changes from previous iterations.

Update management information

Constraints

- ❑ Monitor any changes required in team skills and resources.
- ❑ Monitor any changes to cost and schedule constraints.
- ❑ Monitor any changes to applicable policies.

Update procurement needs and opportunities

Contract vehicles

- ❑ Update and characterize available contracts and procurement vehicles that may be used for needed components and services in the Transition Phase.

Monitor implications of changes to the end user's business process

Organizational change readiness assessment

- ❑ Monitor the behavior of the organization to identify any new reasons for success or failure in changing end-user business processes in support of the solution.
- ❑ Identify any problems or roadblocks to implementing required changes to the end user's business process.

Support requirements

- ❑ Update training requirements for all roles/levels.
- ❑ Update help desk and technical support requirements.

Update risks

Risks

- ❑ Update technical and non-technical risks in the Risk List.
- ❑ Estimate potential risks for the solution that would threaten successful construction and fielding.

Refine the Understanding of the Solution

The solution should be very stable in this phase (if not, the decisions at the LCA milestone should be revisited). The focus of the following tasks is to resolve any unresolved mismatches, to complete any remaining design work, and to design any support materials for initial fielding (beta).

In this phase, any remaining details in the design for the solution are completed. This includes detailed design of end-user business processes, any custom components, and component linkage mechanisms. Mismatches may still be found in this phase. Significant mismatches may arise from changes to components within the solution. Small mismatches may be accommodated if they do not affect the architecture (to include any mechanisms for linking components) or the agreed-upon changes to the end-user business processes. Any changes identified in this phase that affect the definition of the architecture or the end-user business processes should be deferred to a future solution, to maintain the stability necessary to field capability useful to the end users. To build, field, and support a future solution will require repeating all of the EPIC phases.

The tasks below are performed *concurrently* with continuous feedback necessary between the tasks. The tasks do not follow a sequential ordering.

Identify and resolve mismatches from the synthesis of new information

The understanding of the selected components continues to drive the implementation of the solution. The focus of these steps is to assess the impact on the solution of changes to components, existing infrastructure or external interfaces, new end-user business processes, or stakeholder needs. Issues and mismatches are identified for resolution in this or subsequent iterations. In this phase, the solution should be very stable. Occasionally, however, mismatches are significant enough to require that LCA or LCO anchor point decisions be revisited.

The steps below describe the basic steps that must be completed, but the order will be subject to the needs of a particular iteration; they will seldom be implemented in the order shown. In most cases, cycling between the steps will be required as resolution of mismatches in one sphere introduces changes to the baseline and, therefore, potential new mismatches in another sphere.

End-user business processes	❑ Incorporate any new information about end-user or broader organization business processes in the solution. Identify and characterize the nature of any mismatch. Resolve through negotiation, if possible.
Use Cases	❑ Incorporate any new information about Use Cases in the solution. Identify and characterize the nature of any mismatch. Resolve through negotiation, if possible.

Non-functional

- Incorporate any new information about the use-case mechanisms and Supplementary Specification in the solution. Identify and characterize the nature of any mismatch. Resolve through negotiation, if possible.

Architecture

- Incorporate any new information about the architecture in the solution. Identify and characterize the nature of any mismatch. Resolve through negotiation, if possible.

Components

- Incorporate any new information about the components in the solution. Identify and characterize the nature of any mismatch. Resolve through negotiation, if possible.

Cost and schedule

- Incorporate any new information about the cost and schedule of the solution. Identify and characterize the nature of any mismatch. Resolve through negotiation, if possible.

Update the solution if needed

The Solution Vision and the Solution Requirements Specification describe *what* the solution will do. The architecture describes *how* the solution implements the Solution Vision. Expanding the design so that it can be implemented and validated is the primary objective of this set of activities. At this point, the architecture should be stable. However, in the process of accommodating new or changed components or expanding the design, decisions may be made resulting in changes to the specific mechanisms needed to implement the solution.

The steps that follow are performed concurrently with feedback necessary between the steps.

Solution Vision

- Update the features associated with the solution if needed.
  - ❖ The Solution Vision should include a solid understanding of all of the Use Cases.
- Regain agreement with the stakeholders on the Solution Vision if needed.

Requirements

- Update the prioritized stakeholder needs, and update testable functional and non-functional requirements associated with the Use Cases or the Supplementary Specification if needed.

Architecture

- Update the Design Model if needed.

- ❖ Update the external and internal interfaces, if needed.

- ❑ Determine whether to accept new components, releases, or patches.

- ❑ Update the detailed characterization of the interactions of the components if needed.

- ❑ Update the design of any custom components if needed.

- ❑ Update the design of any custom code necessary for integrating the solution if needed.

  - ❖ New component releases will frequently cause new integration strategies to be developed. Each new release should be carefully examined to make sure that the implications of any change are understood.

Waivers

- ❑ Resolve any remaining waivers necessary for "compliance" with the broader organization's architecture or applicable policies.

Architecture Document

- ❑ Update the various high-level architectural views of the system and key decisions and lessons learned.

- ❑ Record agreed-upon changes to the end-user business process.

- ❑ Record approved waivers.

Changes to the end user's business process

- ❑ Update the end-user business processes in the Target Business Use Case and Object Models if needed.

- ❑ Ensure that affected stakeholders understand the scope of changes to the end user's business process and are committed to implementation of the necessary changes.

Training strategy

- ❑ Develop the training strategy for the solution.

  - ❖ The training strategy will have to accommodate many different stakeholder roles. It may be useful to offer tailored training to specific categories of stakeholders.

  - ❖ You may have to consider customizing a vendor's or supplier's standard training to your specific end-user business processes, or provide additional training to complement the

vendor's or supplier's standard training.

- ❑ Ensure that affected stakeholders agree that this strategy will meet their needs.

User support strategy

- ❑ Determine the user support strategy for the solution.
  - ❖ It may be useful to separate the support needed for initial introduction from the long-term support strategy.
- ❑ Ensure the affected stakeholders agree that this strategy will meet their needs.

Cost and schedule

- ❑ Update total ownership costs and schedule for the solution.

Risks

- ❑ Update the risks in implementing the solution.
- ❑ Identify and implement mitigation strategies.

Business Case

- ❑ Update in the Business Case, as needed, the general functionality, performance, quality, fielding approach, changes to the end user's business process, cost/benefit, schedule, and risks of the solution over its anticipated life.

Assemble an Executable Representation

In this phase, all aspects of the solution need to be assembled for fielding to end users. The solution includes the production-quality assembly of the components, any custom code, appropriate linkage to the broader organization's architecture with which the solution must interface, and any changes to the end user's business process necessary to match the processes provided in the components. In addition to assembling a production-quality version of the solution, there is work that must be done to prepare the infrastructure to receive the solution. Work must also be done to implement the organizational changes and business process changes necessary to use the solution.

Furthermore, the Construction Phase includes preparation of the Deployment Artifacts, including installation instructions, version descriptions, user and operator manuals, and other user and installation site support capabilities that are necessary for rollout in the Transition Phase.

Build and test the solution

In this phase, the emphasis is on construction of an Executable Representation that implements the Solution Vision with engineering rigor.

Implementation Model	❑ Define the organization of the Executable Representation in terms of implementation subsystems and/or components organized in layers.
Test Set Artifacts	❑ Complete and update test cases, drivers, stubs, etc.
Implement and test components	❑ Write source code, adapt existing components, compile, link, and execute.
	❑ Submit rework feedback on the design if defects are discovered.
	❑ Tailor and complete testing of the components.
	❑ Fix code defects and perform unit test to verify the change.
	❑ Review the code to ensure quality and ensure coding guidelines are followed.
	❑ Develop integration code or data (wrappers, glue, data sets, etc.) needed to incorporate pre-existing and custom components.
Integrate and test	❑ Integrate new and changed components into a new version.
	❑ Perform integration tests.
Deployment Artifacts	❑ Update the End-user Support Materials to reflect any changes of the Use-case Model in light of any negotiated changes.
	❑ Develop help desk and technical support components.
	❑ Develop user documentation.
	❑ Design or produce training materials.
	❑ Develop the materials needed for long-term support (i.e., Integrated Logistic Support (ILS) of the solution).
Deployment Plan	❑ Implement the Deployment Plan as appropriate.

- ❖ There are multiple views of fielding. In this case, the aspects of the Deployment Plan relating to delivery of the production-quality solution to the end users who will receive the initial fielding should be implemented. This includes end-user support materials, training, etc.

Implement the needed changes to the end user's business process

As the production-quality release is being assembled in the Construction Phase, the end users must prepare to implement the changes to the end user's business process associated with the solution. This is particularly true for those affected by the initial rollout or beta test that will take place in the first iteration of the Transition Phase.

Change management

- ❑ Implement the appropriate elements of the Business Process Change Management Plan.
  - ❖ The implementation must be completed (to the extent possible) for those affected by the initial rollout by the end of this phase.
  - ❖ Review the resistance mitigation strategy.
  - ❖ Revise reward, incentive, and compensation programs.
  - ❖ Restructure the organization if needed.
  - ❖ Establish policies; define standards to support the solution.
  - ❖ Transfer needed knowledge and skills.
  - ❖ Identify solution user group members and develop a charter for the group.
  - ❖ Prototype needed end-user business process change(s) (perhaps using early releases).
  - ❖ Provide the training needed for the end-user business processes associated with the solution.

Make any needed existing infrastructure and external interface changes

The production-quality release is being assembled, and the end users are preparing to implement the changes to the end user's business process associated with the solution release. Make sure the other changes to the infrastructure or operational environment that are needed by the solution proceed in parallel, so that any changes needed in the broader organization's architecture or infrastructure are implemented to support fielding.

Deployment Plan

- Implement the infrastructure elements of the Deployment Plan as appropriate.

Assess the Iteration

As the iteration is completed, it is important to determine if the objectives planned for this iteration were achieved (unresolved issues will be assigned to future iterations). In addition, a review of any unplanned questions, risks, or issues that arose during the iteration must be conducted so that they can be captured in the appropriate planning artifacts. Finally, based on all of the iterations in the phase, a decision is made that the solution is ready for fielding in the Transition Phase.

Assess the Solution

The Executable Representation provides an opportunity for both the end users and the engineers to evaluate the evolving solution in the context of the objectives for the iteration. End users must verify that the business processes represented are acceptable. Engineers must verify that the technical design is implemented correctly. Together they verify that the iteration's objectives have been met, that the evolving solution meets real needs, and that the solution operates in an acceptable manner.

Managing the changes to the end-user business processes requires the same discipline and rigor as does constructing the Executable Representation. As the Business Process Change Management Plan is implemented, the end-user business processes must be assessed to make sure that the end-user business processes described by the solution are acceptable. Use of a preproduction release of the solution may be needed to provide an opportunity for both the end users and the engineers to evaluate the evolving solution in the context of the objectives for the iteration.

- Validate the solution.
  - Show that the solution is what the stakeholders need or want.
- Verify that the solution is implemented correctly.

Update the information about the solution

Screen candidate solutions
- ❑ Update the criteria that were used to select a single solution.

Business Case
- ❑ Amplify the Business Case to capture the significant decisions made in the iteration.

Determine lessons learned from iteration

Iteration Assessment
- ❑ Determine if objectives planned for this iteration were achieved (selected risks mitigated).
  - ❖ Unmet objectives are assigned to future iterations.
- ❑ Identify any unplanned questions, risks, or issues that arose during the iteration and assign them to future iterations.

Risk List
- ❑ Update the Risk List based on the Iteration Assessment.
- ❑ Identify mitigation approaches for the priority risks.

Process improvement
- ❑ Review project metrics and make recommendations for process improvement.

Assess the phase, if the iteration completes the phase

Assessment group
- ❑ Appoint an assessment group with representatives for all affected stakeholders including end users.

Fielding decision
- ❑ Validate that the release ready for use by end users (software solution integrated on the appropriate platforms, current release described, and changes to the end user's business process implemented).
  - ❖ Existing defects and pending changes are not obstacles to achieving the purpose of the next release.

Phase exit criteria
- ❑ The assessment group determines whether the phase objectives and exit criteria have been met and decides whether the project should go ahead. This constitutes the IOC anchor point.
- ❑ The assessment results are documented in the phase Review Record.

Supporting Activities

The activities associated with project monitoring and control, technical process activities, and supporting process activities that were included in the Development Plan have not been described in the activities laid out in this document. They are, however, critical to the success of the project.

Monitor status

Project progress

- Monitor the progress of the project relative to the Project Plan from the viewpoints of the various stakeholders (including budget and schedule).
  - ❖ Include metrics to measure the progress of the solution.
  - ❖ Capture and assess measurements associated with the project's metric goals.
- Manage and control resources and optimize process.
  - ❖ Collect and monitor engineering measures.
- Measure progress in implementing required changes to the end user's business process.
- Monitor and manage resistance of the initial fielding community.

Quality of solution

- Collect and monitor objective measures of the quality of the emerging solution.
- Discover exceptions and problems that must be resolved for project success.

Contracts

- Review and monitor existing contract agreements.
- Monitor any License Agreements for COTS components to ensure they are current and not affected by marketplace changes.

Maintain the experimentation facility

Maintain the experimentation facility in accordance with the Infrastructure Plan to support the identified tasks for this phase and the Transition Phase.

Update and create procurement vehicles as necessary

Appropriate procurement vehicles must be in place to support the ongoing activities in the Construction Phase and the projected activities in the Transition Phase.

Contracts

- Update and create all contracts and procurement vehicles for needed components and services.

License Agreements

- Negotiate licenses if needed.
  - Whereas development licenses will be needed in the Construction Phase, runtime licenses will be needed for the end users in the Transition Phase.

Phase Artifacts

The artifacts from the Construction Phase capture, demonstrate, and verify that

- a production-quality release of the system has been built
- changes to the end user's business process are understood and implemented as appropriate
- the operational environment has been prepared to receive the solution in the Transition Phase

TO CHARACTERIZE END-USER BUSINESS PROCESSES AND STAKEHOLDER NEEDS

Current Business Use-case Model
Current Business Object Model
Target Business Use-case Model
Target Business Object Model
Glossary
Stakeholder Requests
Solution Requirements Specification
Use-case Model
Use Cases
Supplementary Specification

TO CHARACTERIZE THE MARKETPLACE

Market Segment Information
Component Dossier
Component Screening Criteria and Rationale

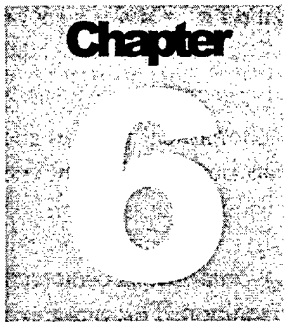
TO CHARACTERIZE THE ARCHITECTURE AND DESIGN

Solution Vision
Design Model
Architecture Document
Executable Representation(s)
▪ Implementation Model (for each Executable Representation)
Test Set Artifacts
Deployment Artifacts
▪ End-user Support Materials

TO CHARACTERIZE PROGRAMMATICS AND RISK

Development Plan (all artifacts are reviewed and updated in each iteration). The artifacts listed below are of particular interest in this phase.
▪ Project Plan
▪ Iteration Plan
▪ Risk Management Plan
▪ Process Improvement Plan
▪ Development Case
▪ Solution Acceptance Plan
▪ Infrastructure Plan
▪ Deployment Plan
▪ Vendor/Supplier Relationship Plan
Business Case

Business Process Change Management Plan
Risk List
License Agreements
Iteration Assessments
Review Record



Transition Phase

The goal of the Transition Phase is to transition the solution to its users. The selected solution is fielded to the user community and supported.

Phase Description

The Transition Phase is focused on moving the solution to the user community. This requires that the users attain proficiency in the solution and the end-user business processes that the solution supports, are motivated to use the solution, and are self-supporting in their use of the solution.

The Transition Phase begins with an initial fielding, or beta test of the solution developed in the Construction Phase. Following a decision to make the solution release generally available, the solution will be fielded across the user base. As required, bugs are fixed, features are adjusted, and missing elements are added to the fielded solution in maintenance releases. Continued monitoring of the marketplace and other sources is required to anticipate changes. Maintaining an experimentation facility for component evaluation to assess the potential impact of any new or changed components is essential.

Following a decision to make the solution release generally available, the solution will be fielded across the user base. Widespread installation and fielding of the solution will often require adaptation to meet the unique needs of specific installation sites. Responsibilities for meeting these unique site needs must be negotiated with the affected site and captured in the Deployment Plan. In addition, during full fielding it is common to experience resistance to implementation of the new capability. This resistance can sometimes be overcome with careful nurturing of champions for the solution among universally regarded experts, and with incentives to reward users for adopting the solution.

The Transition Phase encompasses continued support for the solution. Once the solution has been fielded, the focus shifts to scheduling and implementing maintenance releases of the solution. These maintenance releases can be in response to latent errors in the solution that may require an immediate fix; to the need for enhancements and more routine "bug" fixes that can be accommodated in more routine or periodic releases; or to incorporate

component patches and new component releases. Planning includes determining the errors to be fixed and enhancements to be made in each release. Each maintenance release will require one or more iterations.

Component obsolescence and solution stability must be carefully balanced in support of the solution. This phase manages the introduction of updated components while continuing to meet the demands of the operational environment. Continued monitoring of the marketplace and other sources is required to anticipate changes. Maintaining an experimentation facility for component evaluation to assess the potential impact of new or changed components is essential. In some cases, new component releases may require that the component tailoring and integration (e.g., wrappers, glue) originally used to integrate the components be re-implemented.

At this point in the life cycle, user feedback should focus mainly on fine-tuning the component, configuring, installing and usability issues. All the major structural issues should have been worked out much earlier in the life cycle. The principal artifacts of this phase are

- Executable Representation that is production quality and suitable for release to end users
- Deployment Artifacts that identify changes and known bugs, in a version of a build or unit to be fielded (Release Notes); the software and documented instructions required to install the solution (Installation Artifacts); and material that is used in training programs or courses to assist the end users with solution use, operation, and/or maintenance (Training Materials)
- Market Segment Information and the Component Dossier that supports it

The activities of this phase are required even if support is provided by an organization different from the organization responsible for implementation of the solution. In this case, it is incumbent on the implementation organization to transfer the knowledge that has been gained in the previous phases and iterations to the support organization. Transition of support to a new organization should be considered in planning from the Inception Phase. Transition will affect the amount and kind of information captured in key artifacts and requires the involvement of stakeholders from the support organization starting in the Inception Phase [8].

The Transition Phase ends when the solution is retired and/or replaced by a new solution. A "next-generation" solution requires repeating all of the EPIC phases. It differs from a maintenance release in that the scope of the changes (from stakeholder requests for changes, new or changed components, or new or changed interfaces or linkages to the broader organization's architecture) results in a change in the architecture for the solution, a significant change in the end-user business processes of the end users, or a cost that exceeds the threshold for this phase.

Phase Objectives

Achieve stakeholder concurrence that the baselines for fielding are complete and consistent with the acceptance criteria based on the Solution Vision.

Implement changes to the end user's business process across the user community.

Achieve user satisfaction.

Achieve user self-supportability (e.g., procedures in place, training complete, maintenance plan in place).

Achieve final solution baselines as rapidly and cost effectively as is practical.

Maintain current information about the marketplace and other component sources.

Maintain adequate visibility into component changes through managing relationships with vendors/suppliers.

Balance solution stability with marketplace volatility.

Position procurement vehicles for full fielding and long-term support of the solution.

Collect, analyze, and make accessible for future use information relating to the conduct of the EPIC process.

Phase Task Overview

Phase Planning Activities
- Update the Development Plan for the project

Iteration Activities

Plan the Iteration
- Build a detailed plan for the iteration
- Update the Development Plan for the project

Gather ...

... an Understanding of Stakeholder Needs and End-User Business Processes
- Update and expand the business model as necessary
- Update the behaviors of the solution as needed

... an Understanding of Architecture and Design
- Monitor the architectural context

... an Understanding of Marketplace and Other Sources
- Monitor relevant market segments
- Characterize component changes

... an Understanding of the Programmatics and Risks
- Update management information
- Update procurement needs and opportunities
- Monitor implications of changes to the end user's business process
- Update risks

Refine the Understanding of the Solution
- Identify and resolve mismatches from the synthesis of new information
- Update the solution if needed

Assemble an Executable Representation
- Build and test releases of the solution
- Implement the needed end-user business processes
- Make any needed existing infrastructure and external interfaces changes

Assess the Iteration
- Assess the Solution
- Update the information about the solution
- Determine lessons learned from iteration
- Assess the project if the iteration completes the project

Supporting Activities
- Monitor project status
- Maintain experimentation facility
- Update and create contracting vehicles as necessary

Phase Exit Criteria

Status	Exit Criteria
	Solution functionality is no longer needed or is replaced by a next-generation solution.
	Improvements to, lessons learned, and other useful information on the defined implementation process, tools, and methods are collected and made accessible to future solutions or projects.
	Information is reviewed, analyzed, and used to improve the organization's standard implementation process.

Phase Planning Activities

Update the Development Plan for the project

The Development Plan is updated in each iteration based on new information and lessons learned in engineering and management activities in previous iterations. The Development Plan references the entire set of major planning artifacts that describe the project and how it will be executed. In this phase the plans necessary to field the solution are completed. Iterations in this phase focus on establishing the solution as the operational baseline for all users. The focus then shifts to ongoing support of the solution until it is retired or replaced.

Development Plan

- ❑ Review the project organization. Verify that the level of staff resources and skills available are appropriate.
- ❑ Update the Project Plan to include scheduling for anticipated releases.
  - ❖ Plan the work to modify and reintegrate the solution.
- ❑ Review the plans for project monitoring and control and update as needed.
- ❑ Update the Risk Management Plan.
- ❑ Review the plans for technical process planning and update as required. In particular, update the plan for maintaining the experimentation facility in the Infrastructure Plan.
- ❑ Review the Vendor/Supplier Relationship Plan; ensure that information channels appropriate to the importance of each major vendor/supplier are maintained.
- ❑ Review the plans for the other supporting processes and update as needed.

Test planning

- ❑ Update the Solution Acceptance Plan if needed. Validate any changes with stakeholders.
- ❑ Review and update, as necessary, the test objectives and plan in the Test Set Artifacts, in support of new solution releases.

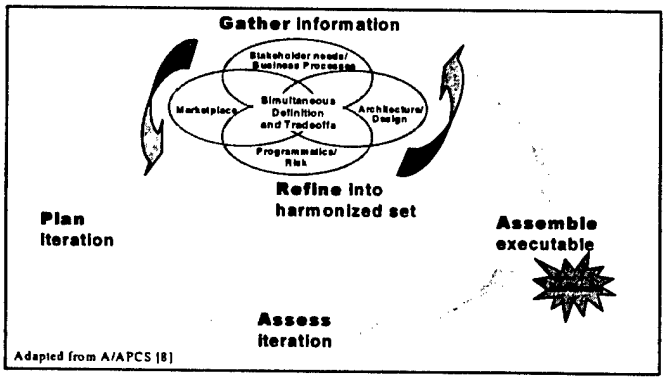
Plan for fielding

- ❑ Update and validate with end users the Deployment Plan, including plans for data migration for each release of the solution.

End-user business process change planning	❑ Update and validate with end users a detailed Business Process Change Management Plan for needed changes to organizational structure and/or end-user business processes. This should match the Deployment Plan for each solution Release.

Iteration Activities

ITERATION ACTIVITIES KEY
Plan
Gather
Refine
Assemble
Assess



Plan the Iteration

Plan the iteration in detail at the beginning of each iteration. The focus of the iterations in the Transition Phase is to build production-quality maintenance releases of the solution. Any lessons learned and new risks discovered from previous iterations are incorporated in the Risk List that is a key consideration for defining the iteration objectives and activities. The information (and the level of detail of that information) to be **gathered** about each of the spheres of influence is determined by the iteration objectives. The iteration objectives also determine what analysis must be done to **refine** the information from each of the spheres. Of more importance in this phase, the iteration objectives determine the content of the Executable Representations to be **assembled**.

Build a detailed plan for the iteration

The detailed plan for the iteration is based on the current Risk List and the most critical functionality. Risk is a key discriminator in deriving objectives for the iteration. The highest priority risks are mitigated as early as possible. However, addressing risk is balanced with ensuring that the critical functions and services the solution provides are addressed early.

Iteration Plan

- Refine the scope of the iteration and the goals and objectives that were planned in the Project Plan for this iteration to reflect any changes since the plan was last updated.
- Define objectives for the success of the iteration.
  - These objectives will provide focus for all activities in the iteration and will be used at the end of the iteration to decide whether the iteration was a success.

- Define the requirements for the Executable Representation needed to demonstrate the iteration objectives.
  - This *evolutionary* prototype is created in addition to one or more *exploratory*, throwaway prototypes to mitigate specific risks such as design and requirements tradeoffs, component feasibility analyses, or demonstrations of key scenarios to certain stakeholders conducted as part of **gather** or **refine**.
  - Depending on the specific objectives of the iteration, more than one prototype may be needed
- Identify the tasks that will be required to achieve the iteration objectives and the specific artifacts that must be developed or updated.
- Complete a detailed work breakdown structure to show how the work that must be done within the iteration is allocated and what resources are necessary to do the work.
- Determine milestones (events and dates) that are important to the iteration.

Update the Development Plan for the project

Development Plan

- Revise and update the Development Plan.
  - The plan should be updated to reflect changes to the technical or programmatic baseline, to reflect changes in personnel availability or skills, to reflect the changes necessary to accommodate a particular set of components, or simply to reflect a new approach to meeting the identified needs.

Gather Information

The **gather** activities in this phase are oriented to monitoring the four spheres for any changes that affect the fielded solution. The information gathered from one sphere continues to depend on and drive the information needed from another sphere. Therefore, once a change has been identified in one sphere, it is likely that the **refine** tasks for the iteration will require additional **gather** activities to collect complementary information from the other spheres of influence to meet the iteration objectives. The **gather** and **refine** activities will continue to cycle until the information becomes sufficiently detailed to meet iteration objectives.

Gather an Understanding of Stakeholder Needs and End-User Business Processes

This phase monitors changes in the organization's business model that could affect release and fielding. In addition, lessons learned from use of the solution in operations are collected to capture errors in implementing the Use Cases and stakeholder requests for solution enhancements to be considered in this release or future maintenance releases.

Update and expand the business model as necessary

There should be very few, and only minor, changes to the business model at this point. Pay particular attention in this phase to the site-specific variations to the business use cases to ensure they are adequately covered in the Deployment Plan.

Mission

- ☐ Review and update the understanding of the mission and strategic direction of the end-user organization as well as the forecast of potential changes and forces of change that will affect the end-user organization and its mission over time.
- ☐ Review and update any business goals of the end-user organization that relate to, or are affected by, the solution.

Stakeholders

- ☐ Review any changes in stakeholders identified with a vested interest in the outcome of the solution.

Current fielding capabilities

- ☐ Review and update, as necessary, the characterization of the current operational environment for each of the relevant stakeholders who will use some aspect of the solution.
  - ❖ The emphasis is on site-specific variations that affect fielding.

- ❑ Verify that current fielding capabilities are understood as well as any methods and resources currently in place to implement system upgrades or major enhancements.

Business measures

- ❑ Review and update as necessary the business-related measurement goals and the metrics that will be used to evaluate the solution.

Business model

- ❑ Review and update the Current Business Use-case Model and Current Business Object Model.
  - ❖ The emphasis is on site-specific variations that affect fielding.
- ❑ Review and update the Target Business Use-case Model and the Target Business Object Model.
- ❑ Update the Glossary.

Update the behaviors of the solution as needed

The characterizations of needs for the solution should only change slightly based on operational experience with the solution. Changes may be necessary if there are significant component changes during the Phase. Maintain agreement with the stakeholders and set realistic expectations on what changes will be made.

Stakeholder Requests

- ❑ Review and update Stakeholder Requests as necessary.

Use-case Model

- ❑ Review and update the complete set of Use Cases; validate that the right set has been captured, revisit the prioritization of the Use Cases.

Use Cases

- ❑ Review and update the Use Cases as necessary.

Use-case mechanisms

- ❑ Review and update the mechanisms and services that are needed by the Use Cases.

Supplementary Specification

- ❑ Review and update the non-functional needs, quality attributes, and design constraints in the Supplemental Specification.

Gather an Understanding of Architecture and Design

In the Transition Phase, this **gather** activity monitors the broader organization's architecture and external systems for changes that may affect the solution. In addition, this **gather** activity monitors the implementation of the solution for architectural and design lessons learned implications.

Monitor the architectural context

Architecture

- Review and update the details about the architecture constraints, boundaries, and interfaces (including interfaces to external systems and existing infrastructure).

Gather an Understanding of Industry and Marketplace

In this phase it continues to be important to monitor the marketplace for changes–either in the structure or direction of the market, or in new or changed components. Any new or changed components identified will be examined in the experimentation facility to determine how they might impact the release that is being fielded. The decision to change components or upgrade component versions, and the determination as to when to make such changes, will have to be carefully balanced against the need to maintain a stable operating environment for the end users.

Monitor relevant market segments

Look for any indications that may affect the vendor's long-term support of the components used by the solution (technology maturity, component obsolescence, component splits or mergers, vendor's going out of business, buy-outs, etc). In addition, look for new components that may drive the definition of the next generation solution and work with other customers to gain leverage over vendors in defining and delivering the changes, patches, and components needed.

Market Segment Information

- Monitor the technologies and components used in the solution for any changes or forecasts of change.
- Monitor the market for new components that may be relevant to the solution.
- Monitor the buying habits of the other buyers, the delivery of component releases, the behavior of the vendors/suppliers, and size and distribution of each relevant market segment to track and forecast trends.
- Update the relevant Market Segment Information.

Characterize component changes

In this phase there will likely be changes to the selected components. In addition, new components may have been introduced in the marketplace. A copy of each component, release, or patch should be obtained and loaded in the experimentation facility for detailed examination. Both end users and engineers should examine each changed component to understand how any changes might affect the solution.

Component capabilities	❑ Characterize the capabilities and limitations of component changes in the Component Dossier. ❖ It is important to make sure that the component still does what is needed in supporting both the end-user business processes and the architecture and that there are no unintended side affects from the changes. ❑ Evaluate the integration impacts of any component changes. Identify any architectural or end-user business process impacts that result from the change.
Licenses	❑ Monitor and update licensing options information for COTS components.
Vendor/supplier drivers and health	❑ Monitor the vendor/supplier's market strategy, general release frequency, typical relationships with buyers, typical licensing arrangements, and long-term viability. Update the Component Dossier as needed.
Screen candidate components	❑ Screen newly identified components against Component Screening Criteria and Rationale; capture the rationale for removing any components from further consideration in the Component Dossier.

Gather an Understanding of the Programmatics and Risks

It remains important to monitor changes in information developed and maintained external to the project (i.e., policy directives, budget, and sometimes schedule and/or priorities), as well as to capture the programmatic information changes from previous iterations.

Update management information

Constraints

- ❑ Monitor any changes required in team skills and resources.
- ❑ Monitor any changes to cost and schedule constraints.
- ❑ Monitor any changes to applicable policies.

Update procurement needs and opportunities

Contract vehicles

- ❑ Update and characterize available contracts and procurement vehicles that may be used for needed components and services.

Monitor implications of changes to the end user's business process

Business Process Change Management Plan

- ❑ Identify lessons learned from initial fielding (e.g., from changes to the end user's business process, technology insertion approaches).
- ❑ Identify any problems or roadblocks to implementing required changes to the end user's business process.

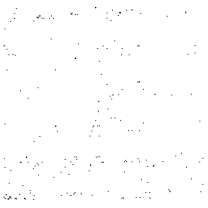
Support requirements

- ❑ Identify any additional training requirements for all roles/levels.
- ❑ Monitor help desk and technical support requirements.

Update risks

Risks

- ❑ Update technical and non-technical risks in the Risk List.
- ❑ Estimate potential risks for the solution that would threaten successful fielding and support.



Refine the Understanding of the Solution

In this phase, the solution is adjusted to accommodate fixes to identified prioritized problems and minor enhancements (if not, the decisions at the IOC milestone should be revisited). The focus of these activities is to identify and resolve any mismatches and to complete any design work necessary to support a fieldable release.

Small mismatches may be accommodated in the detailed design if they do not affect the architecture (to include any mechanisms for linking components). Any significant changes, however, should be deferred to a future solution to maintain the stability necessary to field capability that is useful to the end users and establishes a foundation for building future solutions.

The tasks below are performed *concurrently*, with continuous feedback necessary between the tasks. The tasks do not follow a sequential ordering.

Identify and resolve mismatches from the synthesis of new information

These activities assess the impact to the solution of changes to components, the broader organization's architecture or external interfaces, new end-user business processes, or stakeholder needs. Issues and mismatches are identified for resolution in this or subsequent iterations.

The steps below describe the basic steps that must be completed, but the order will be subject to the needs of a particular iteration; they will seldom be implemented in the order shown. In most cases, cycling between the steps will be required as resolution of mismatches in one sphere introduces changes to the baseline and, therefore, potential new mismatches in another sphere.

End-user business processes	❑ Incorporate any new information about end-user or broader organization business processes in the solution. Identify and characterize the nature of any mismatch. Resolve through negotiation, if possible.
Use Cases	❑ Incorporate any new information about Use Cases in the solution. Identify and characterize the nature of any mismatch. Resolve through negotiation, if possible.
Non-functional	❑ Incorporate any new information about the use-case mechanisms and Supplementary Specification in the solution. Identify and characterize the nature of any mismatch. Resolve through negotiation, if possible.
Architecture	❑ Incorporate any new information about the architecture in the solution. Identify and characterize the nature of any mismatch. Resolve through negotiation, if possible.

Components

- Incorporate any new information about the components in the solution. Identify and characterize the nature of any mismatch. Resolve through negotiation, if possible.

Cost and schedule

- Incorporate any new information about the cost and schedule of the solution. Identify and characterize the nature of any mismatch. Resolve through negotiation, if possible.

Update the solution if needed

The Solution Vision and the Solution Requirements Specification describe *what* the solution will do. The architecture describes *how* the solution implements the Solution Vision. The emphasis in this phase is to assess the impact of changes to the solution baseline. Adjusting the design so that changes can be evaluated, implemented, and validated in **assemble** and **assess**, respectively, is the primary objective. In the process of adjusting the design to accommodate fixes and enhancements, decisions will be made that may change the specific mechanisms needed to implement the solution. If this is the case, changes should be considered for the next solution rather than a maintenance release.

The steps that follow are performed concurrently, with feedback necessary between the steps.

Solution Vision

- Review and update the features associated with the solution if needed.
- Maintain agreement with the stakeholders on the Solution Vision.

Requirements

- Review the prioritized stakeholder needs for enhancements. Update testable functional and non-functional requirements associated with the Use Cases or the Supplementary Specification if needed.

Design Model

- Update the Design Model if needed.
  - ❖ Update the external and internal interfaces in the Design Model if needed.
- Determine whether to accept new components, releases, or patches.
- Update the underlying mechanisms (or utilities) to support the implementation of the Use Cases if needed.

- ❑ Update the detailed characterization of the interactions of the components if needed.
- ❑ Update the design of any changed custom components or new code necessary for integrating the solution.

Waivers

- ❑ Resolve any new waivers necessary for "compliance" with the broader organization's architecture or other applicable policies.

Architecture Document

- ❑ Update the various high-level architectural views of the system and key decisions and lessons learned.
- ❑ Record agreed-upon changes to the end user's business process.
- ❑ Record approved waivers.

Changes to the end user's business process

- ❑ Update the end-user business processes in the Target Business Use Case and Object Models if needed.
- ❑ Ensure that affected stakeholders understand the scope of changes to the end user's business process and are committed to implementation of the necessary changes.

Training strategy

- ❑ Update the training strategy for the solution release if needed.
  - ❖ You may have to consider customizing a vendor's or supplier's standard training to your specific end-user business processes, or provide additional training to complement the vendor's or supplier's standard training.
- ❑ Ensure that affected stakeholders agree that this strategy will meet their needs.

User support strategy

- ❑ Update the user support strategy for the solution release if needed.
  - ❖ It may be useful to separate the support needed for initial introduction from the long-term support strategy.
- ❑ Ensure that affected stakeholders agree that this strategy will meet their needs.

Cost and schedule

- ❑ Update total ownership costs and schedule for the solution.

Risks

- Update the risks in implementing the release of the solution.
- Identify and implement mitigation strategies.

Business Case

- Update the Business Case as needed.

Assemble an Executable Representation

All aspects of the solution need to be assembled for fielding to end users in each release. The solution includes a production-quality assembly of the components; any custom code; appropriate linkage to the broader organization's architecture with which the solution must interface; and any changes to the end user's business process necessary to match the processes provided in the components. In addition to assembling a production-quality version of the solution, work may need to be done to prepare the infrastructure to receive the solution and to implement the organizational changes and business process changes necessary to use the solution.

The Executable Representation in this phase corrects defects and adds minor enhancements. In addition, the Transition Phase includes completion of the Deployment Artifacts, including installation instructions, version descriptions, user and operator manuals, and other user and installation site support capabilities that are necessary for fielding and long-term support.

Build and test releases of the solution

In this phase, the emphasis is on implementing an Executable Representation that implements the Solution Vision by making small changes to the solution with the engineering rigor necessary for releases that will be fielded.

Implementation Model

- Update the organization of the Executable Representation in terms of implementation subsystems and/or components organized in layers as needed.

Test Set Artifacts

- Update test cases, drivers, stubs, etc.
  - ❖ Use any reported bugs to enhance the test cases.

Implement and test components

- Write any new source code; update existing components; compile, link, and execute.
- Submit rework feedback on the design if defects are discovered.

- Tailor and complete testing of the components as needed.
- Fix code defects and perform unit test to verify the change.
- Review the code to ensure quality and ensure coding guidelines are followed.
- Update integration code or data (including wrappers, glue, data sets, etc.) needed to incorporate components and custom components.

Integrate and test

- Integrate new and changed components into a new version.
- Perform integration tests.

Deployment Artifacts

- Update the End-user Support Materials to reflect the negotiated changes.
- Update the help desk and technical support components.
- Update user documentation.
- Update training materials.
- Update the materials needed for long-term support (i.e., Integrated Logistic Support (ILS) of the solution).

Deployment Plan

- Update and (re)implement the Deployment Plan as appropriate.

Implement the needed end-user business processes

As the production-quality release is being assembled, the end users must prepare to implement the changes to the end user's business process associated with the solution release.

Change management

- Implement the appropriate elements of the Business Process Change Management Plan.
  - ❖ Review resistance mitigation strategy. (Resistance can sometimes be overcome with careful nurturing of champions for the solution among universally regarded experts)
  - ❖ Revise reward, incentive, and compensation programs.

- ❖ Restructure the organization if needed.
- ❖ Update policies and standards to support the solution.
- ❖ Transfer needed knowledge and skills.
- ❖ Expand solution user groups.
- ❖ Implement the new end-user business processes (reorganization, training, data migration) for full fielding.
- ❖ Provide training needed for the end-user business processes associated with the solution.

Make any needed existing infrastructure and external interface changes

The production-quality release is being assembled, and the end users are preparing to implement the changes to the end user's business process associated with the solution release. Make sure the other changes to the infrastructure or operational environment that are needed by the solution proceed in parallel, so that any changes needed in the broader organization's architecture or infrastructure are implemented to support fielding.

Deployment Plan

- ☐ Implement the infrastructure elements of the Deployment Plan as appropriate.

Assess the Iteration

As the iteration completes, it is important to determine if the objectives planned for this iteration were achieved (unresolved issues will be assigned to future iterations). In particular, it must be decided whether to accept new component releases. In addition, a review of any unplanned questions, risks, or issues that arose during the iteration must be conducted so that they can be captured in the appropriate planning artifacts. Finally, a determination is made as to whether the new solution release is ready for fielding.

Assess the Solution

The Executable Representation provides an opportunity for both the end users and the engineers to evaluate the new solution release in the context of the objectives for the iteration and to validate the baselines against the Solution Vision using the criteria in the Solution Acceptance Plan. End users must verify that intended fixes and enhancements operate as expected and that associated changes to the end user's business process are acceptable. Engineers must verify that the fix or enhancement is implemented correctly and does not introduce new problems. Together they verify that the iteration's objectives have been met and that the solution release meets real needs, and operates in an acceptable manner.

Managing the changes to the end user's business processes requires the same discipline and rigor as constructing the Executable Representation. As the Business Process Change Management Plan is implemented, the end-user business processes must be assessed to make sure that the end-user business processes described by the solution are acceptable. Use of a preproduction release of the solution may be needed to provide an opportunity for both the end users and the engineers to evaluate the evolving solution in the context of the objectives for the iteration.

- Validate the solution.
  - ❖ Show that the solution is what the stakeholders need or want.
- Verify that the solution is implemented correctly.

Update the information about the solution

Screen candidate solutions
- Update the criteria that were used to select the solution.

Business Case
- Amplify the Business Case to capture the significant decisions made in the iteration.

Determine lessons learned from iteration

Iteration Assessment
- Determine if objectives for this iteration were achieved.
  - ❖ Unmet objectives are assigned to future iterations or the next solution.
- Identify any unplanned questions, risks, or issues that arose during the iteration and assign to future iterations.

Risk List
- Update the Risk List based on the Iteration Assessment.
- Identify mitigation approaches for the priority risks.

Fielding decision
- Validate that the release is ready for use by end users (software solution integrated on the appropriate platforms, current release described, changes to the end user's business process implemented).
  - ❖ Existing defects and pending changes are not obstacles to achieving the purpose of the next release.

Process improvement
- Review metrics and make recommendations for process improvement.

Assess the project if the iteration completes the project

At the end of the project, it is important to look back across all of the iterations to collect and analyze lessons learned so appropriate process changes can be made for future projects.

Assessment group	❑ Appoint an assessment group with representatives for all affected stakeholders including vendors/suppliers.
Project closeout	❑ Ensure that the stakeholders agree that functionality in the solution is retired or replaced (i.e., that there is no longer any requirement for support to this solution).
	❑ Settle the project finances.
	❑ Archive all project documentation and records.
	❑ Transfer the project measurements to the corporate historical database.
	❑ Reassign remaining project staff.
Lessons learned	❑ Conduct a review of the lessons learned from the project. Capture the results in a Review Record.
	❑ Update the process artifacts to incorporate the lessons learned.

Supporting Activities

The activities associated with project monitoring and control, technical process activities, and supporting process activities that were included in the Development Plan have not been described in this document. They are, however, critical to the success of the project.

Monitor project status

Project progress

- Monitor the progress of the project relative to the Project Plan from the viewpoints of the various stakeholders (including budget and schedule).
  - This includes measuring the progress of the solution.
  - Capture and assess any measurements associated with the project's metrics goals.
- Manage and control resources and optimize processes.
  - Collect and monitor engineering measures.
- Monitor implementation of the Business Process Change Management Plan and identify corrections as necessary.
- Measure progress in implementing required changes to the end user's business process.
- Monitor and manage resistance to full implementation of the solution in the end-user community.

Quality of solution

- Collect and monitor objective measures of the quality of each solution release.
  - Objective measures will include factors such as performance, downtime, user satisfaction, and productivity improvements.
- Discover exceptions and problems that must be resolved for project success.

Maintain the experimentation facility

Maintain the experimentation facility in accordance with the Infrastructure Plan to support the identified tasks for this phase.

Update and create contracting vehicles as necessary

Contracts

- Review and monitor existing contract agreements. Update as needed.
- Monitor and update any License Agreements for COTS components to make sure they are current and not affected by changes in the marketplace.

Phase Artifacts

The artifacts from the Transition Phase capture, demonstrate, and validate that

- Production-quality maintenance releases of the system have been built that fix errors and provide minor enhancements to the solution.
- Changes to the end user's business process are understood and implemented as appropriate.
- The operational environment has been prepared to receive the solution release.

TO CHARACTERIZE END-USER BUSINESS PROCESSES AND STAKEHOLDER NEEDS

Current Business Use-case Model
Current Business Object Model
Target Business Use-case Model
Target Business Object Model
Glossary
Stakeholder Requests
Solution Requirements Specification
Use-case Model
Use Cases
Supplementary Specification

TO CHARACTERIZE THE MARKETPLACE

Market Segment Information
Component Dossier
Component Screening Criteria and Rationale

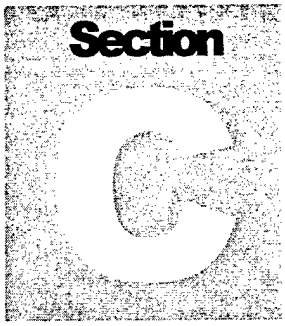
TO CHARACTERIZE THE ARCHITECTURE AND DESIGN

Solution Vision
Design Model
Architecture Document
Executable Representation(s)
▪ Implementation Model (for each Executable Representation)
Test Set Artifacts
Deployment Artifacts
▪ End-user Support Materials
▪ Release Notes
▪ Training Materials
▪ Installation Artifacts

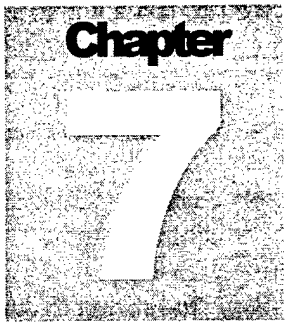
TO CHARACTERIZE PROGRAMMATICS AND RISK

Development Plan (all artifacts are reviewed and updated in each iteration.) The artifacts listed below are of particular interest in this phase.
▪ Project Plan
▪ Iteration Plan
▪ Risk Management Plan
▪ Process Improvement Plan
▪ Development Case
▪ Solution Acceptance Plan
▪ Infrastructure Plan

▪ Deployment Plan
▪ Vendor/Supplier Relationship Plan
Business Case
Business Process Change Management Plan
Risk List
License Agreements
Iteration Assessment (one/iteration)
Review Record



Guidelines and Artifacts



Market Segment Information Guidelines[15]

Overview

The Marketplace Segment Information artifact captures the broad characteristics of the market represented by a set of competing components that are under consideration for use in the system. These characteristics include vendors, suppliers and buyers participating in the market, COTS components offered, processes automated, technologies represented, procurement strategies practiced, and competitive market forces. The focus of the market segment information artifact is on large-scale market dynamics rather than in-depth analysis of individual components.

Purpose of Captured Information

The market segment information artifact accumulates and organizes information sufficient to

- verify that the general marketplace represented by the COTS components is appropriate for the project (e.g., appropriately stable, innovative, customer-centered, organized, etc.)
- verify that active buyers and users of the capabilities produced in the marketplace have needs similar to those of the project
- verify that active buyers and users of the capabilities produced in the marketplace are employing marketplace offerings in a manner similar to that anticipated by the project
- identify the strategies used by active buyers and users for acquiring components produced by the marketplace
- identify the competitive pressures that drive market and component improvements and changes, marketing strategies, etc.

[15] The information in this guideline is adapted from http://www.cadv.org/cadv.htm

- verify that COTS components can be appropriately placed in the system
- establish the veracity of one or more potential solutions incorporating COTS components

Key Questions to Answer

- What is the relationship of the quantity we intend to buy vis-à-vis the quantities that others buy? Will our volume justify a lower-than-market price due to the seller's increased economies of scale? Will our volume be so large as to drive the sellers to or beyond full capacity, resulting in unanticipated inflation?
- What are the pricing strategies of firms in the market? What are the implications for expected prices?
- Is there a cyclical pattern to supply and demand? What forces might drive prices in the near future? Strikes? Labor shortages? Subcontractor bottlenecks? Energy shortages? Other raw material shortages?
- What features distinguish one component from another? What is the apparent tradeoff between features and price?
- What are the commercial warranty terms and conditions (if any)? What are the historical repair costs for each component? What are the historical maintenance costs for each component?
- What terms and conditions are used in commercial transactions? What terms and conditions have been used in other government acquisitions? What type of contract is generally used in commercial transactions? Government acquisitions?
- What has been the historical default rate by firms performing similar contracts? What performance problems have typically been encountered?

Information Needed

Competitive Market Forces

- ☐ Identify the level of market competition and the number of potential sources capable of satisfying the stakeholder needs.
  - ❖ Include the status, size, and location of sellers in the market.
- ☐ Identify market prices and pricing trends.
- ☐ Identify customary terms and conditions governing commercial sales of the component.

- ❖ Include factors that affect market prices.
- ❖ Include market information on dollar and unit amounts of component sales.
- ❖ Include trends in buying practices.
- ❖ Include business practices peculiar to the marketplace.
- ❖ Include trends in commercial and government sales.

- ❑ Identify barriers to new firms entering the marketplace.
- ❑ Identify business growth, expansions, and declines.

Buyers, procurement practices and approaches

- ❑ Identify active component buyers and users.
  - ❖ Include characteristics (size, function, nationality, etc.) of active buyers and users of components in the market.
  - ❖ Include business processes supported by buyers and users in the market.
  - ❖ Include similarity and differences between these business processes and those required.
  - ❖ Include market technologies and components employed by buyers and users.
  - ❖ Include related technologies employed by buyers and users.
  - ❖ Include average dollars spent and other resources expended.
  - ❖ Include evidence of success or failure of systems incorporating market components.

- Identify potential market growth and size.
  - Include growth forecasts.
  - Include barriers to market growth.
  - Include common procurement strategies for buyers and users comparable to this organization.
  - Include common implementation strategies of comparable buyers and users.
  - Include implementation status for comparable buyers and users.

Applicable industry-wide laws or regulations

- Identify pending legislation or regulation.
- Identify antitrust or competitive practice litigation.
- Identify reports regarding safety problems and fraud in the marketplace.

Market consolidation

- Identify buyouts and mergers.

Market standardization

- Identify standardization of functional capabilities, technologies, and practices.
- Identify bodies governing vendor practices and component capabilities.
- Identify consortia and other organizations dedicated to collaboration between vendors and components.
- Identify cooperative agreements.

Technological changes and trends

- Identify basic technologies competing in the marketplace (e.g., object technology, relational database technology).
- Identify relative maturity of basic technologies.
- Identify trends in marketplace acceptance of common technologies.
- Identify components representing competing technologies.

	❑ Identify emerging technologies.
	❑ Identify relative maturity of emerging technologies.
	❑ Identify market penetration for emerging technologies.
	❑ Identify projected market penetration for emerging technologies.
Component changes and trends	❑ Identify new components in development for future availability.
	❑ Identify technologies employed in new components.
	❑ Identify functional changes expected in new components.
	❑ Identify cost/complexity of upgrade from current to future components.

Techniques

Techniques[16] for conducting market research may include any or all of the following:

- contacting knowledgeable individuals in government and industry regarding market capabilities to meet stakeholder needs
- reviewing the results of recent market research undertaken to meet similar or identical stakeholder needs
- publishing formal "requests for information" or "sources sought" synopses in appropriate technical or scientific journals or business publications
- querying government databases that provide information relevant to agency acquisitions
- participating in interactive, on-line communication among industry, acquisition personnel, and customers
- obtaining source lists of similar items from other contracting activities or agencies, trade associations, or other sources
- reviewing catalogs, periodicals and other generally available literature published by manufacturers, distributors, and dealers or available on-line

[16] Adapted from the Federal Acquisition Regulations (FAR) Part 10

- reviewing professional journals including those specific to a segment of the marketplace
- conducting interchange meetings or holding presolicitation conferences to involve potential offerors early in the acquisition process

In addition, the following should be considered:

- attending sales presentations, trade shows, fairs, and symposia
- attending professional associations meetings and conferences
- participation in relevant consortia
- telephone or mail surveys of buyers in the market
- direct contacts with customers referenced by the contractor
- conducting studies, surveys, and interviews with industry, state, or local governments
- phone surveys or formal questionnaires/surveys
- customer references to enable the agency to evaluate others' experiences with the component and their observations regarding its quality, reliability, and maintainability
- unsolicited comments regarding previous procurements, including comments made during pre- or post-award debriefings of unsuccessful offerors

Methods

The Commercial Advocates Forum, an Internet source whose URL is http://www.cadv.org/cadv.htm, provides access to I-Mart, a comprehensive search tool for locating potential sources. I-Mart searches are based on a description of the component or service, Federal Supply Classification (FSC), or Federal Supply Group (FSG). It utilizes various search engines that can be selected to search for sources by industry. I-Mart can be contacted at http://www.imart.org.

Sample Survey Vehicles

Acquisition History

Desired Information	Response
Name of the contractor who received previous award, and award date	
Description of supply/service	
How well did the component or service meet the needs of the requiring activity?	
What was the cost (award price and delivery terms)?	
Was the item competitive?	
What types of problems were encountered?	
What method was used to procure the requirement?	
What type of contract was used?	
Were there any unsolicited comments and complaints regarding previous procurements?	
What are the results of any synopses?	

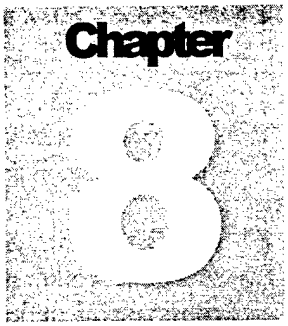
Buyer Market Survey

Desired Information	Response
What component(s) do you currently use to meet your needs?	
Who are your suppliers?	
What is the cost?	
What is the normal delivery time?	
Was performance satisfactory? If not, explain.	

Is the component commercial? If not, explain.	
What type of discounts does supplier offer, if any?	
What are supplier's warranty terms, if any?	
Is there any additional charge for special packing and packaging?	
Are there any recommendations?	

Industry Component Survey

Desired Information	Response
What type of components do you offer to meet specified stakeholder needs?	
Did you have difficulty with previous specifications?	
Do you offer a commercial component?	
Is there a better method of soliciting for these stakeholder needs?	
Are the components still available?	
Have the components been changed? If so, how often?	
What type of discounts do you offer your customers?	
What are your warranty terms, if any?	
Is there any additional charge for special packing and packaging?	
Do you have a commercial catalog?	
What is your average delivery lead time?	
Is there an alternative component that would meet the same need?	



Component Dossier Guidelines

Overview

The Component Dossier artifact captures all of the information regarding a single COTS component. Information captured in the Component Dossier includes characteristics of the vendor, component architecture and functional capabilities, standards supported, required hardware and software configurations, non-functional characteristics like usability, supportability, reliability, interoperability, portability, and scalability, and quality of documentation, costs, and licenses. A Component Dossier is initiated for each component considered during the project. A Component Dossier on a component discovered in the Inception Phase is expanded during the Elaboration Phase as the component is further considered (and potentially selected), and is kept current (for example, reflecting additional information discovered about a component or new component version updates) during the Construction and Transition Phases. A component discovered after the Inception Phase may still be considered for use in the solution. A Component Dossier is created when the component is introduced and updated as appropriate.

Purpose of Captured Information

The Component Dossier artifact accumulates and organizes information sufficient to

- record the history of contacts with the vendor regarding the component
- record the history of consideration and use of the component
- record raw (unfiltered) information about a component and component vendor gathered directly from the vendor (documentation, claims, price lists, demonstration versions, etc.), and from third parties (such correspondence and reviews by other users, trade journal articles, business/financial analysis, etc.)
- record processed (filtered) data obtained during consideration of a component including the results of investigations into the component and vendor, information

describing the exact configuration of the component evaluated, and data gathered during evaluation activities and benchmarking

- record the analysis of the component and vendor, including component/vendor strengths, weaknesses, related components and ensembles, and architecture or usage constraints identified during evaluation
- record the history, rationale, and specific activities for customization and tailoring of the component
- record the history, rationale, and specific activities for integration of the component
- record the history of version releases
- record the history and rationale for upgrade decisions and certification activities

Information Needed

The goal for populating the Component Dossier is to capture information sufficient to select (or rule out) use of a specific component version; to maintain data about the architectural, design, implementation, and testing ramifications of use of the component; to transition necessary skills to stakeholders (such as maintainers and end users); and to support the maintenance/evolution process of the component in the system.

The categories of information maintained within the Component Dossier are extensive. Some of this information is developed to support the selection of the component. Other information is developed as the component is incorporated and maintained in the system. Thus, a Component Dossier is a living document that represents the state of knowledge about a component during the time it is considered for, used in, and maintained for the system. Examples of the categories of information that are maintained in a Component Dossier are identified below. The type and degree of information maintained for each category will depend on a number of factors, including the characteristics of the component, the stage in component selection and use, and how the component is or will be used in the system. In addition to example categories, sample questions that illuminate the intent of the categories are provided.

Vendor characteristics

Organizational stability	❑ Has the organization existed in its present form for a suitable period to indicate that it is stable?
Financial stability	❑ Is the organization making money?
	❑ What are the financial trends?

Nationality	❑ Is the organization based in the U.S. or a nation allied with the U.S.?
Ease of access	❑ Is there sufficient access to the organization for answering technical and business questions?
Independence	❑ Does the vendor make independent decisions, or is it (effectively) controlled by another organization?
	❑ Are the goals and directions of the controlling organization appropriate for the needs of the target system?
Reputation	❑ Does the organization have a reputation for quality?
	❑ Is delivery timely?
	❑ Is it responsive to customers?
Support infrastructure	❑ Does the organization offer local offices, hotlines, installation and integration support, etc.?
Engineering approach	❑ Is the engineering approach used by the organization appropriate and compatible with my organization's expectations and needs?
Maintenance approach	❑ Is the maintenance approach appropriate and compatible?
History	❑ What is the history of the organization? Where did it come from, and how did it get to the position of marketing this component?

Basic component characteristics

Shipment dates	❑ When was the component first made available to customers?
Component Stability	❑ What is the release history of the component?
	❑ What types of changes were made for various releases?

Install base	❑ How many copies of the component are in use?
	❑ How many organizations use the component?
	❑ Are these organizations similar to the target organization?
	❑ Can the use of the component by these organizations be verified (i.e., not marketing hype or shelfware)?
Customer references	❑ What customer references are available?
	❑ How do these customers use the component, when did they take delivery, how many copies of the component do they use, and how many users are supported?
	❑ What are their impressions of the vendor, component, support, etc.?
	❑ Is the use of the component by these customers similar to the anticipated use of the target organization?
End of life plans	❑ What phase-out or end-of-life planning is the vendor for the component considering?
	❑ When is phase out or end of life planned?
	❑ What will the upgrade path be?
	❑ What will this upgrade require of users?
	❑ Are any plans documented and available to customers?
Availability of training	❑ What training is available for the component, when and where is it offered, and is it tailored to the customers' needs?
	❑ For what groups of stakeholders (system personnel, maintainers, end users, etc.) is training available?
	❑ Are there any third parties providing training?
Access to hotline	❑ During what hours of operation is a hotline available?
	❑ What types of support are available?

	❑ Are hotline calls fielded domestically?
	❑ Are there appropriate capabilities to maintain required security?
Consultants	❑ Are vendor-sanctioned consultants available?
	❑ Are third-party consultants available?
	❑ What is the availability and cost for consulting?
Delivery method	❑ What media is used for delivery of the component and component upgrades (tape, CD, Internet, etc.)?

Standards

DoD standards	❑ What DoD-specific standards are supported?
Industry standards	❑ What general industry standards are supported?
	❑ What standards body is responsible for the standard?
	❑ How do organizations join or influence the direction of the standard?
	❑ Is the standard widely supported?
	❑ Does one (or a few) organization have extensive control over the standard?
	❑ What is the release history for the standard?
	❑ How can contact be made with the group or committee responsible for the standard?
Organizational	❑ Do the component and vendor meet special standards, procedures, and protocols required by target organization?
Completeness	❑ Does the component implement a subset of the standard, the complete standard, or a superset of the standard?
	❑ What are the plans for update or enhancement to subsequent versions of the standard?
Confidence	❑ How is standards compliance verified?

Hardware

Configuration	❑ What is the minimal hardware configuration (computers, processors, memory, disk, bus, peripherals, etc.), recommended configuration, and maximum configuration?
	❑ What incremental steps can be made in hardware to increase performance, storage capacity of the system?
	❑ Does the required hardware configuration conflict with that of any other system with which the component must interact or be collocated?
	❑ Is a special or different engineering, testing, or support environment required?
Communications	❑ What communications infrastructure is required?
	❑ What bandwidth?
	❑ What configuration?
Hardware compatibility	❑ Are there any known compatibility problems between the component and hardware components?
Accuracy	❑ Is the accuracy of all hardware components within the required configuration appropriate for my organization's needs?
Security	❑ Is the security of all hardware components within the required configuration appropriate for my organization's needs?
Reliability	❑ Is the reliability of all hardware components within the required configuration appropriate for my organization's needs?
Vendor characteristics	❑ Are vendor characteristics for all hardware components within the required configuration appropriate for my organization's needs?
Component characteristics	❑ Are the characteristics for all hardware components within the required configuration appropriate for my organization's needs?

Upgrade	❑ How is the upgrade of a hardware component tied to upgrade of the component? ❑ How long after upgrade of hardware is a component upgrade generally available? ❑ How long are old versions of hardware supported by the component?

Software

Operating system	❑ What operating system(s) are required (including versions)? ❑ Are the performance and size characteristics appropriate for the needs of the target system? ❑ What mechanisms exist to identify and resolve problems related to the interface between the operating system and the component? ❑ Who is responsible for identifying and resolving the problem?
Communications	❑ What communications support is required (including versions)? ❑ Are alternative communications capabilities supported? ❑ Are the performance and size characteristics appropriate for the needs of the target system? ❑ What mechanisms exist to identify and resolve problems related to the interface between communications capability and the component? ❑ Who is responsible for identifying and resolving the problem?
Database	❑ What database support is required (including versions)? Are alternative databases supported? ❑ Are the performance and size characteristics of the supported database(s) appropriate for the needs of the target system?

	❑ What mechanisms exist to identify and resolve problems related to the interface between the database and the component?
	❑ Who is responsible for identifying and resolving the problem?
Related applications	❑ What other applications are required (including versions)?
	❑ Are there alternates for these applications?
	❑ Are the performance and size characteristics appropriate for the needs of the target system?
	❑ What mechanisms exist to identify and resolve problems related to the interface between the related applications and the component?
	❑ Who is responsible for identifying and resolving the problem?
Compatibility problems	❑ Are there any known compatibility problems between the component and any software component?
Accuracy	❑ Is the accuracy of all software components within the required configuration appropriate for the needs of the target system?
Security	❑ Is the security of all software components within the required configuration appropriate for the needs of the target system?
Reliability	❑ Is the reliability of all software components within the required configuration appropriate for the needs of the target system?
Vendor characteristics	❑ Are vendor characteristics for all software components within the required configuration appropriate for the needs of the target system?
Component characteristics	❑ Are the characteristics for all software components within the required configuration appropriate for the needs of the target system?

Upgrade	❑ How is the upgrade of a software component tied to upgrade of the component?
	❑ How long after upgrade of software is a component upgrade generally available?
	❑ How long are old versions of software supported by the vendor?

Usability

Intended use and users	❑ Who are the intended users of the component?
	❑ For what use was it intended?
General operability	❑ How hard is the component to use?
Skill level required	❑ What skills are required by users?
Responsiveness	❑ What is the response time under light load? Average load? Peak load?
	❑ Can response times be tuned or improved?
Robustness	❑ What is Mean Time Between Failure for the component?
	❑ How does the component respond to erroneous input and operator error?
Help capabilities	❑ What help capabilities are available in the component?
Error assist/recovery	❑ How does the component assist users when they make an error in input of data?
	❑ How does the component support users in recovery from erroneous input?
Understandability	❑ Is the component easy to understand?
	❑ Are common usage paradigms employed?
Learnability	❑ How long will it take before employees will be proficient with the component?

Supportability

Dependencies	❑ Does the component make use of any component or capability provided by an organization other than the vendor?
	❑ To what extent does success of the component within the target system depend on these organizations?
	❑ How is failure of a component produced by another party handled?
	❑ How would subcontractors fare if subjected to the same evaluation scrutiny as the vendor?
Upward compatibility	❑ Have all versions of the component been upward compatible?
	❑ Which versions have not been and why?
	❑ What steps must be taken when a new release of a component must be installed?
Site installation support	❑ Who is responsible for installation of the component on-site?
	❑ Will the vendor install the component?
	❑ Is there extra cost for this service?
	❑ Can target organization personnel install the component?
	❑ What skills are required?
Site operation support	❑ Will the vendor provide personnel to support initial operations, perform standard maintenance, or diagnose errors?
	❑ Does the component indicate to users/operators when maintenance is necessary or an error has occurred?
Analyzability	❑ Does the component provide capabilities to analyze performance?
	❑ To locate problems or bugs?
	❑ If capabilities are not provided, how is this accomplished?

Replaceability	❑ If the component must be replaced with another commercial component, what changes would be necessary to the system?
	❑ What activities would be necessary for data migration?
Preventive maintenance	❑ Is periodic preventative maintenance required?
	❑ How frequently?
	❑ What activities are involved?
Special support	❑ Is a special or different engineering, testing, or support environment required?
	❑ What are the characteristics and components of that environment?
	❑ What tools are required or suggested?

Interoperability

Data model/format	❑ What data model and formats does the component employ?
	❑ Are they published?
	❑ What standard are they based on?
	❑ What other components support the same data model/formats?
Support for data access	❑ What interfaces or techniques are available to access component data?
	❑ What effort is required to access component data?
	❑ Is the granularity of data access appropriate for the target system?
Support for control	❑ Can the component be invoked by other applications? How?
	❑ At what granularity can the component be invoked?
	❑ Can other components control low-level functions that might be necessary in the integrated system (for example, commit for a change)?

- ❑ Can the component invoke other applications? How?
- ❑ What constraints are placed on these invocations?
- ❑ How can execution of the component and other components be synchronized?
- ❑ What timing concerns may arise?

Infrastructure utilized

- ❑ What infrastructure is used to support communications of messages, data, and control sequencing within the component?
- ❑ Can the infrastructure be used by other system components to interact with the component?

Reliability

Test regimen

- ❑ How does the vendor perform testing?
- ❑ Are the results of testing independently verified?
- ❑ Are test scripts and results available?

Type/frequency of faults

- ❑ What is the mean time between failures?
- ❑ What is the frequency of different sorts of faults?

Recovery from faults

- ❑ What is the error handling strategy?
- ❑ Is there journaling of faults?
- ❑ Are all faults trapped before the system panics?

Benchmarking

- ❑ Are reliability benchmarks available for the component?
- ❑ Are any claims made about reliability?

Experience

- ❑ Do systems requiring similar reliability to that of the target system use the component?
- ❑ Which ones?

Performance

Benchmarking	❑ Are performance benchmarks available for this component?
	❑ Are the results of these benchmarks suitable?
	❑ Do the benchmarks reflect a usage situation or pattern consistent with that expected of the component in the target system?
Time-related behavior	❑ Does the component exhibit appropriate time-related behavior (throughput, lack of deadlock, thread-safety, latency, etc.)?
	❑ Is there any potential for time-related interactions with other system components? Where?
	❑ Have these interactions been evaluated and determined to be within acceptable limits or risk levels?
Resource behavior	❑ Does the component make appropriate use of resources (processors, memory, devices, etc.)?
	❑ Is there a possibility of contention for resources with other system components?
	❑ Have these contentions been evaluated and determined to be within acceptable limits or risk levels?
Surge capacity	❑ Does the component have the capability to handle increasing loads as expected (e.g., increased number of transactions, increased complexity of processing, increased number of tracks, etc.)?
Adaptability/flexibility	❑ Can the component be tailored to efficiently handle an appropriate range of performance expectations (transaction rates, numbers of tracks, etc.)?
	❑ How is this adaptation accomplished?

Documentation

Design information	❑ Is available design information sufficient to determine whether the design is appropriate?
	❑ Is it sufficient for determining an integration strategy with other target system components?
Maintenance information	❑ Is the available maintenance information sufficient for installation?
	❑ for routine use?
	❑ for preventative maintenance?
	❑ For fault isolation and recovery?
Training materials	❑ Are training materials and courses available?
	❑ Are they appropriate?
	❑ Are they affordable?
	❑ Do they cover an appropriate set of stakeholders for the target system?
	❑ Are training material/courses tailored for specific stakeholders?
Customization	❑ Can documentation, training materials, design information, maintenance information, etc., be customized for unique target system needs?
	❑ What is involved in customization?
	❑ What will it cost?
Quality	❑ Is the quality of all documentation and other information appropriate?
Policy on reproduction	❑ Can materials be reproduced as needed?

Licenses

Usage/maintenance	❑ Are standard usage maintenance licenses appropriate for the target system?
	❑ Are license terms negotiable?
	❑ Is site licensing and/or quantity discounting available?
Transferability of license	❑ Are licenses transferable to other operating units or other agents working on behalf of the target organization?
RT licensing	❑ Are separate licenses necessary/available for development and fielded platforms?
	❑ What are the terms of these licenses?
Data rights	❑ What data rights are included in the standard license?
	❑ Are these appropriate for the target system?
	❑ Must additional data rights be negotiated?
Escrow	❑ Can source code be escrowed?
	❑ What are the costs and stipulations of that escrow?
	❑ Is an escrow a reasonable precaution for this system?
Discontinuation	❑ What rights does the target organization have if the component is discontinued?
Expiration	❑ What events occur when a license expires?
	❑ Is there any notification of impending expiration?
	❑ Are licenses "time bombed?"

Functional capabilities

Appropriateness	❑ Does the component offer appropriate functional capability?
	❑ Is this functionality provided in an appropriate manner (appropriate process, interfaces, quality, etc.)?
Process consistency	❑ Are the processes supported by the component appropriate to our organization?
	❑ Which of our internal processes must change?
	❑ How will this change be accomplished?
Industry practices	❑ Does the component conform to best industry practice?
	❑ How was this determined?
Completeness	❑ What proportion of the intended system capability does the component provide? How was this determined?
	❑ What is the mismatch between the functions necessary in the target system and those supported by the component?
	❑ What level of effort will be required to provide missing capabilities or enhance deficient capabilities? How should this be accomplished?
Tailoring/customization	❑ Is the component suitable "out of the box" or does it require custom construction of scripts, code, tables, etc.?
	❑ What effort is involved in performing this customization? Who will perform this customization?
	❑ Must this effort be repeated to incorporate new component releases?
Excess	❑ Does the component offer additional functional capability that will not be used? Should not be used?
	❑ What impact does this additional capability have on resource stakeholder needs, performance, etc.?

Architecture

Component	❑ What architectural paradigms are evident in or employed by the component?
	❑ Are they appropriate for the target system?
System	❑ Does the component suggest architectural paradigms for the target system?
	❑ Does the component impose architectural restrictions on the system? Are these appropriate?
	❑ What is the impact on other system components?

Component Version Data

Version ID	❑ What are the version number and release date of the component?
	❑ What additional information is needed to uniquely identify the component (e.g., revision number, patch number, etc.)
Version documentation	❑ Identify all component documentation, including end-user support materials, reference manuals, release notes, installation instructions, known bug lists, etc.
Version capabilities	❑ What new features, capabilities, and fixes are provided by this uniquely identified component?

Component/System Relationship

System configuration	❑ What system configurations does the component work with (or is part of)?
System adaptation	❑ What environment variable settings are required?
	❑ What specific settings are required for networking, memory, processes, peripheral devices, etc.?
	❑ What adaptation and settings are required of other components of the system to work with this component?

Integration

- What (new) assumptions or expectations does the unique component version make regarding interaction with other components in the environment?
- What changes must be made to the assumptions made by the rest of the system regarding the behavior of this version?
- What integration guidelines must be followed and specific integration activities undertaken?

Tailoring/modification

- What tailoring or modification of the component is required?
- What settings are required for component variables?
- What scripts, tables, schemas, 4GL code, etc. are required? Why are these required?
- Were workarounds considered? Why were they rejected?
- Has the tailoring/modification been approved by an authoritative control board?
- Was the component vendor consulted? What was the vendor's response?
- Will tailoring/modification affect the contract in any way (e.g., changes in license fees, changes in maintenance practices or responsibilities)?
- What assurance is there that the modified version will become part of the standard commercial offering?
- Who has performed or will perform the tailoring/ modification?
- Are all applicable test data and verification of test passage under configuration control?

Component Dossier Artifact

Description

The Component Dossier Artifact can best be understood as an index that identifies and locates all of the information that represents the current understanding of a component. This information is produced and stored in different formats. Some of it (for example, contact information for the vendor) may be physically stored as part of the Component Dossier. Other information may be indexed by the Component Dossier but stored elsewhere. For example, the executable for a component may be represented in a tape library, component documentation may be at a network address, and reports produced by the vendor or the organization considering the component may be represented in a file cabinet. The purpose of the Component Dossier is to tie all of these divergent artifacts that represent a single instance of a component and use of that component together into a logical unit.

Template

1.0 Component Identification

- 1.1 Name
- 1.2 Version number (rev number, patches installed, etc.)

2.0 Vendor Contact Information

3.0 Component Description

[summary of what the component does and what it is being considered for/how it is used in the system]

4.0 Component Status

[current state of decisions made regarding use of the component, whether it has been selected, is being used, actively maintained, or being replace/retired]

5.0 State of Evaluation, Testing, Certification

6.0 Vendor Data (includes both raw and processed information)

- 6.1 Financial
- 6.2 Business
- 6.3 Engineering

7.0 Component Data (includes raw and processed information)

- 7.1 Basic Characteristics
- 7.2 Standards
- 7.3 Hardware/Software Configuration Required

7.4 Functional Capabilities

7.5 Non-functional Capabilities
[usability, supportability, interoperability, reliability, security, etc.]

7.6 Performance

7.7 Documentation

7.8 Licensing

7.9 Architecture

8.0 System Relationships, Tailoring, and Modifications (includes raw and processed information)

8.1 System Configuration

8.2 System Adaptation

8.3 System Integration

8.4 Component and System Tailoring and Modification

9.0 Component Usage History

9.1 Dates considered, used, retired

9.2 Bugs/problems reported

9.3 Disposition of bugs/problems

9.4 Queries to vendor or third parties for support

9.5 Changes/updates to configurations and tailoring
[capture rationale, changes, and results]

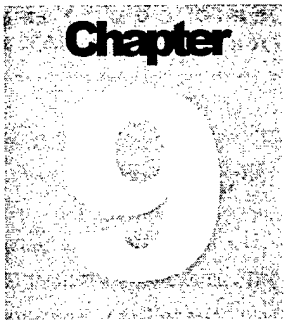
9.6 Preventative/other maintenance performed

10.0 Dossier Usage History

10.1 Who, what, and why record of access to Dossier components

10.2 Errata or inconsistencies found

10.3 Additional information required



Component Screening Criteria and Rationale Guidelines

Overview

The Component Screening Criteria and Rationale artifact documents the set of criteria against which components are initially evaluated and, ultimately, accepted or rejected for consideration as part of a solution. The artifact is initiated during the Inception Phase as solution acceptance criteria are formed and are amplified as candidate solutions incorporating COTS components are formulated. The Component Screening Criteria and Rationale artifact is revised during other phases as solutions are refined and new solutions (and components) are considered.

In theory, it may be necessary to create one Component Screening Criteria and Rationale artifact per solution considered, because the implications of the solution on system capability, architecture, and design may affect the expectations placed on COTS components. In practice, the criteria used to screen COTS components differ only slightly across multiple solutions for the same system, since the core system expectations remain constant. Thus, it is usually adequate to create a single Component Screening Criteria and Rationale artifact that describes both the consistent screening requirements and the divergent requirements based on specific solutions.

Purpose

The Component Screening Criteria and Rationale artifact accumulates and organizes information sufficient to

- document preliminary expectations regarding vendor, functional, operational, architectural, market placement, and supplemental characteristics that must be present for a component to be considered as part of a solution
- document rationale describing why these expectations are sufficiently critical to require all candidate components to comply and how they will distinguish between candidates
- identify measurable criteria that correspond to preliminary expectations
- document a data gathering strategy and any specific processes and techniques required to determine whether a component meets measurable criteria
- capture, in summary form, any activities that result in a component being eliminated from further consideration

Discussion

The first step in developing an initial Component Screening Guideline and Rationale is to identify the set of critical expectations regarding COTS components. Significant information about component expectations will have been developed in parallel to construction of the Market Segment Information artifact. Information that leads to the investigation of one market segment over another begins to define initial screening criteria. Information regarding features that distinguish one component from another and competitive pressures that drive vendors to evolve components will prove particularly useful. Additional distinguishing features of COTS components and vendors may be derived from the categories of information identified for the Component Dossier.

When this information is vetted in light of the developing business models, critical use cases, architectural context, and procurement strategies for the system, an initial set of critical expectations for individual COTS components will normally emerge. As the understanding of a specific solution advances, unique expectations for COTS components may be uncovered and should be incorporated into the Component Screening and Rationale.

The step of converting from loosely formed expectations to highly structured component screening criteria involves identifying expectations that are

- highly discriminating (i.e., that remove components from future consideration)
- readily translatable into measurements
- quickly and economically evaluated

Organizations often use functional requirements of a system as component screening criteria. One advantage of the use of functional criteria is that they are directly measurable–a COTS component either provides the function required or not. However, one goal of the EPIC process is to avoid overly constraining technical solutions and COTS selections early in the development process. If this goal is to be met, functional requirements must be specified with course granularity during early phases of the process. Unfortunately, it is often hard to distinguish between major marketplace competitors based on course functions (e.g., virtually all accounting packages maintain a general ledger). Even if requirements are detailed such that discriminating criteria can be identified, gathering necessary information about individual components can require extensive hands-on effort or access to expensive experts.

One approach that can be helpful in identifying appropriate functional screening criteria involves the application of specialized tools that are intended to assist in component selection for specific application domains (e.g., ERP). Such tools are available directly from vendors (e.g., SoftSelect for the ERP domain) or as part of services offered by large engineering and/or consulting houses (e.g., Gartner Group). The risk with use of such tools is twofold: they can lead to specification of requirements at more detailed levels than appropriate for early stages of development, and, while focusing on functional capabilities, they overlook other non-functional expectations (e.g., vendor characteristics, market placement data, component architecture) for selected COTS components.

Since many COTS component failures are attributed to deficits in non-functional characteristics like reliability, security, and maintainability, it is tempting to use these "quality attributes" as screening criteria. However, gathering good information on these characteristics is particularly hard without extensive hands-on experience. Anecdotal information, whether provided by vendors, customers, or a third party, can be suspect. Information gathered from these sources is likely to express some bias. It also does not take into account the particular expectations for the system under development.

In contrast, vendor characteristic and market placement data are eminently available, tend to be impartial, and by nature are highly discriminating. This is particularly true for well-developed markets where there are a few top vendors followed by a larger number of niche players. Vendor and market characteristics (e.g., vendor size, market share) readily distinguish among vendors–and are often indicators of more refined characteristics that are critical for DoD application (ability of a vendor to manage a large contract, provide a wide range of services, and remain viable over the long term). While care must be taken not to unduly eliminate niche players that cater to a particular market (such as the DoD), it is hard to argue against applying screening criteria like vendor size and market share when addressing very large systems. Additional particularly useful vendor characteristic and market placement screens include

- nationality–domestic components are often preferred for legal and security reasons
- stability–due to the long lifetime of DoD systems, stability of the vendor and product are paramount

Where system architecture is partially defined by existing components or high-level architecture guidelines, criteria focused on the high-level architecture of COTS components are particularly fruitful for initial screening. Architectural information is readily available from vendor web sites, and often indicates whether a component supports specific technologies (e.g., Java 2 Platform, Enterprise Edition). However, it is important to verify this information before it is accepted as fact. Useful architectural screening criteria include

- low-level interoperability to platforms, databases, etc.
- matches to already-selected components involving infrastructure, build processes, control models, data manipulation models, etc.)
- matches to already-selected components involving the patterns of interaction identified by protocols and characteristics of the data communicated
- assumptions about presence or absence of other system components.

Tips

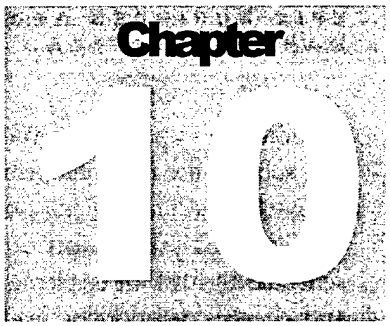
Strike a reasoned balance between efficiency and completeness

Efficient screening often requires employing high-level criteria that can quickly exclude many COTS components. For example, a common and often appropriate screening strategy suggests that examining only the top few (e.g., 3-5) competitive components in terms of sales or customer-installed base. However, there are often reasons to additionally consider smaller, niche players that provide a unique or tailored capability more in line with system expectations. This suggests that there may be multiple pathways for inclusion of a component.

Avoid premature focus on detailed architecture and design

While high-level architecture is an appropriate screening criteria, there is a tendency to prematurely focus on detailed architecture and design as screening criteria (for example, specifying the architectural characteristics unique to one component) Where architectural or design decisions have already been made that **must** be reflected by the chosen component, then it is important that criteria reflect these decisions. However, to fight against the tendency to think within the box, consider the risks and potential workarounds if a highly similar capability is delivered in a different manner.

Optimize the order in which you apply screening criteria	Criteria will vary along dimensions of ability to discriminate and effort required to obtain data. By their nature, some criteria will be highly discriminatory in determining which components are appropriate. Other criteria, while important, will eliminate fewer components from consideration. Still other criteria will require more work than the norm to accumulate data. Consider both how discriminating a criterion is and the effort to evaluate components against the criterion when determining an order for applying screening criteria.
Market share/growth are good screens for long-lived systems	A defining characteristic of many systems is expected lifespan. Often systems are used for over 20 years. COTS components in these systems should come from financially sound companies with good prospects to remain in business.



Business Process Change Management Plan Guidelines[17]

Purpose

In a world of rapidly changing products and punishing competition, mastering the management of business process changes can be a key competency that distinguishes winners from losers. (Approximately 70% of business process reengineering efforts, or redesigns, fail [20,21].) The adoption of a powerful concept, process, method, and/or tool often holds the promise of dramatic benefits to an organization. Efforts to realize these benefits, however, often result in frustration and anger from precisely those people who should benefit from the adoption. Business process change management is, therefore, not an isolated activity but a process that touches many of the socio-technical activities at work in an organization. A structured approach to managing the human elements is critical to achieving strategic business objectives.

Gaining predictable value from a new technology solution may require a number of people in the organization to do their jobs differently. End-user business processes may change specifically to meet business objectives or they may change to accommodate the business processes inherent in components under consideration within a technical solution. An integral part of building and fielding a new technology solution requires anyone affected by

[17] Adapted from works of the Accelerating Software Technology Adoption (ASTA) Initiative of the Software Engineering Institute, Carnegie Mellon University, PA. *Technology Adoption–an Overview* (draft paper) by Carter, L. and *Organizational Change Readiness Assessment* (draft paper) by Carter L. and Defaud, S. and *Managing Software Technology Transition as a Project* by Fowler, P.

the solution to participate in defining and evaluating the solution so that they understand and support the tradeoffs between business process changes required in various aspects of their jobs and an evolving understanding of the capabilities of the components available "off-the-shelf", the design of the architecture that links the components, and the costs, schedule and risk of implementing the solution. This support must survive any turnovers and changes in priorities, leadership, management, and reorganizations that occur in the organization from the time the change is discovered and when it is fully implemented across the organization.

All approaches to change involve assumptions and expectations about what is central in a change effort. Many see "new technologies dominating the improvement effort; others focus on changes in process or business practices; some perceive organizational change. Very few, however perceive change in multiple facets of the organization (process, technology, people, culture, organization)"[22]. SEI studies have found that successful business process changes tend to be those where the organization has the following skills:

- sponsorship development and sustainment: generating and maintaining informed and effective sponsorship to bring about a significant change and lead the effort
- improvement assessment: assessing current workflows, concentrating on discovering a new work flow (way of accomplishing the required result by doing the work differently and better
- technology evaluation: evaluating technology alternatives and selecting the best fit for the desired new work flow
- solution tailoring: tailoring the technology and skill development materials to fit the culture and needs of the adopting organization
- usage design: designing work and management activities to support the new workflow and producing training, skill development, and behavior change aids to support those whose jobs will change as a result of the adoption (both technical and managerial)
- adoption planning: planning and staging the adoption to maximize the benefit to the adopting groups at times that gracefully fit with other commitments and the availability of resources
- adoption implementation: implementing the adoption as a mission-critical project with involved and proactive leadership
- skill development: developing the skills required by both the people who will use the technology and those overseeing these people
- pilot testing: validating the adoption approach and materials are effective in this organization
- organizational roll-out: refining the adoption approach and materials based on usage experience and as required for each new organizational unit, phasing the adoption implementation to gracefully fit each unit's point in their respective project life cycle

- lessons learned: learning from each adoption experience and feeding these lessons forward into future efforts

The IDEAL℠ Model

The IDEAL℠ model is an organizational improvement model that serves as a roadmap for initiating, planning, and implementing organizational improvement actions. The IDEAL model is named for the five phases it describes: initiating, diagnosing, establishing, acting, and learning. The IDEAL model provides a structure that focuses on a number of critical elements that must take place in managing a successful business process change. The table below captures the 5 phases and the major activities within each phase in the IDEAL model.

Initiating	Laying the groundwork for successful improvement effort ▪ Set context ▪ Build sponsorship ▪ Charter infrastructure
Diagnosing	Determining where you are relative to where you want to be ▪ Characterize current and desired states ▪ Develop recommendations
Establishing	Planning the specifics of how you will reach your destination ▪ Set priorities ▪ Develop approach ▪ Plan actions
Acting	Doing the work according to the plan ▪ Create solution ▪ Pilot/test solution ▪ Refine solution ▪ Implement solution
Learning	Learning from the experience and improving your ability to adopt new technologies for the future ▪ Analyze and validate ▪ Propose future actions

℠ IDEAL is a service mark of Carnegie Mellon University. Additional information about IDEAL can be found on the SEI web site <http: www.sei.cmu.edu>.

Implementing Business Process Change Management

The quality and appropriateness of a new technical solution has been shown to be a poor predictor of the success of a fielding in terms of the actual utilization of the new useful capabilities. A one-size-fits-all approach to fielding and utilization of a solution seldom works. Successful fielding and utilization requires an appreciation of the unique characteristics of each organization into which the technical solution will be fielded, and activities to support utilization will be tailored to fit those characteristics. These characteristics will be examined in the context of the IDEAL model.

Initiating

Most organizations wish to adopt a new technology solution to bring about significant improvement with respect to some measure of performance. Such significant improvements will require dramatic changes to the way work is performed in the organization (or there would be no significant improvement), which means that many people will have to change the way they do their jobs. The initiating phase comprises those activities aimed at gaining the commitment of the organization's leadership and supporting infrastructures to take ownership of, responsibility for, and proactive leadership of the adoption effort. Involving all of the relevant stakeholders is critical, for they may have other potentially conflicting efforts in progress and success can only be realized with their acceptance of the adoption.

Sponsorship development and sustainment

Without informed and active leadership, few business process change efforts will be successful [23]. The dramatic changes that must take place will result in near-term decreasing in organizational effectiveness before the benefits are realized [24]. Without the informed support of the organization's leader, most efforts fail to survive this decline. The business process change effort will require time from initiation to completion, so effort to sustain the informed sponsorship is also critical. It is not uncommon for sponsorship to be withdrawn. Those charged with the success of the adoption must strive to ensure that the sponsor fully appreciates the benefits and the costs and does not lose sight of the goal during the process.

One of the first issues that must be addressed in building sponsorship is the mission need/benefit that motivates the needed business process change. What is the truly compelling mission need that drives the solution? If there is no obvious answer to this question, why proceed? Similarly, if the realistic costs associated with the solution exceed the potential benefit, why proceed? Few business process changes driven by "it's the right thing to do" are successful if a compelling motivation is absent.

The sponsor must have adequate power and influence to place the organization in a position to absorb some near-term pain for the sake of long-term mission success. If there truly is a mission need for change and the change is not trivial, then some critical resources will have to be pulled from performance of today's operational mission. It takes real leadership to recognize when and how to sacrifice to ensure the mission can always be

performed. Without the constant support of a powerful leader, the inevitable crisis of the moment will always derail strategic initiatives. For this reason, the adoption team must establish and then diligently sustain sponsorship for the adoption. Getting the initial blessing to proceed is important, but retaining it is fundamental.

Diagnosing

Having the will to improve (sponsorship) is just the beginning of the process. Knowing what to change and why is the next step. Adoption of a new technology solution does not automatically result in an improvement. Without understanding how the organization performs its work, where bottlenecks exist, and where effort is expended, it is possible to adopt technology and damage the organization's capacity to perform its mission. The goal of the diagnosing phase is to understand what isn't working well and to suggest ways to improve.

There are many different assessment methods aimed at addressing many different issues. The Baldrige Assessment [25] provides insight into how well the organization's management has established the basic processes for forming a group of people into an effective team for performing its mission of providing quality products and services to satisfied customers. The CMM[SM] Based Appraisal for Internal Process Improvement [26] is used for helping software engineering organizations surface issues that may damage their software capability and for providing a stimulus for improvement. A Software Risk Evaluation [15] is a method for considering potential problem areas by means of leveraging the experiences of numerous professionals who have contributed to the taxonomy that underlies the method. Even an ad hoc assessment by an experienced professional (usually external to the organization) can be quite beneficial in surfacing issues.

It is critical not to lose sight of the goal of an assessment. An assessment is intended to surface problems. If the sponsor has not been properly engaged, the findings from the assessment can be threatening and can lead to an untimely termination of the adoption effort. The goal of an assessment is to understand the current state of the organization in an honest, unbiased manner, so efforts to improve it by means of technology adoption can be established in a meaningful way. Implementation of the plan and realizing the new capabilities of the solution depend on a set of individuals, respected by the organization and its leadership, who are authorized, supported, equipped, skilled, and dedicated to the success of the change.

The following components are designed to collect information about the target organization with respect to the implementation effort, assemble the data, and build an implementation plan that will increase the likelihood of success.

Improvement assessment

Realizing the new capabilities of a technology solution begins with an assessment of the size of the change implied by the new solution. Assessing how the organization currently performs its work and appreciating where delays and needless effort occur is fundamental. Planning the implementation of a change benefits from having a list of all of the people who

must change their jobs and knowing how their jobs must change. Without this knowledge, business process changes are just as likely to damage as improve the organization. Before selecting a technology solution, the organization must determine what needs to be corrected and why. The clarity of this insight and the ability to communicate it effectively (without assigning blame) can be a powerful tool in keeping sponsorship for the effort alive and well.

The team will use the information gathered on the current state as they define the technical solution and rollout approach. Any new business process associated with technology adoption is modeled as part of one or more iteration's Executable Representation to focus analysis on the most important features, to consolidate the views of the key stakeholders, and to document requirements. In this stage, the team

- defines the requirements for managing the business process change (later refined to be detailed, verifiable, focused, and clarified)
- performs baselining activities to get a picture of the organization's current strengths and weaknesses in the technology area
- analyzes improvement opportunities and drafts a new business process
- prototypes a whole-product solution (at a high level)

Basic questions are

1. What is the nature of the capability being transitioned?
2. What is the state of the organization that will incorporate the new technology solution?
3. What is the ultimate goal for acquiring and using the technology?
4. What are the steps to reach the desired goal, given the state of the organization, including its human capabilities?

Attributes of the information needed are

- relative advantage. The degree to which an innovation is perceived as being better than its precursor.
- compatibility. The degree to which an innovation is perceived as being consistent with the existing values, needs, and past experiences of potential adopters.
- complexity. The degree to which an innovation is perceived as being difficult to use.
- observability. The degree to which the results of an innovation are observable to others.
- trialability. The degree to which an innovation may be experimented with before adoption.
- image. The degree to which use of an innovation is perceived to enhance one's image or status in one's social system.
- acceptance. The degree to which use of the innovation is perceived as being voluntary.

Organizational Readiness

An assessment of the organization's readiness to implement the change and having insight into the readiness of the organization to implement the change is an essential element of the Business Process Change Management Plan. An effective change plan typically tailors the elements of the solution to fit the organization and its specific needs, leverages the organization's strengths, addresses the barriers pointed out by the readiness assessment, and ensures that each critical person in the organization has the understanding, knowledge, skill, motivation, and support needed to be successful.

A readiness assessment along the following six dimensions can be performed several ways (questionnaires, interviews, etc.), but there are advantages and disadvantages to each. It is for this reason that the use of an experienced professional is recommended. Producing effective questionnaires, conducting interviews that do not color the results, interpreting the responses, and producing effective change plans is not something that people without proven skill can accomplish without considerable coaching.

1. **What is leadership's commitment to the technology solution? What is their willingness and capacity to**
   - actively lead the utilization of the solution through
     - ownership of the technology solution and its processes
     - communication of context, rationale, setting unique and inspiring vision
   - change the way they perform and talk about their leadership role in implementing the solution
   - take heat during efforts to become skilled in the use of the technology solution when it appears that things are going slowly or when some tactical crises appears including
     - perseverance to stay the course within the leadership team
     - building contingency strategies
   - supply the **right** resources (as opposed to available resources)
   - create commitment infrastructure that includes
     - performance-based reinforcement from all management levels
     - rewards for those willing to step up to the changes required by the solution
     - sanctions for those who resist the changes required by the solution (even when they try to hide behind some mission-critical tasking)
   - put things into place to make it difficult for their successor to undo the change to the solution without first understanding its rationale, the investment being made, and the consequences of terminating the change. Potentially successful approaches include
     - installing content bottom-up; implementing top-down
     - feedback loops instituted throughout all levels to ensure unfiltered feedback
     - meaningful monitoring mechanisms

2. **What is the history of similar changes in the organization in the past? Does leadership**
   - ❑ acknowledge past failures and understand why these changes failed, including
     - ❖ understanding what didn't work and is likely not to work again
     - ❖ demonstrating why this effort is different and why it will succeed
   - ❑ create a sense of urgency by actions such as
     - ❖ demonstrating visible, painful changes on the part of the leader
     - ❖ creating quick, visible success effort
   - ❑ honor the lessons that have been learned and deserve to be honored
   - ❑ demonstrate the inevitability of this implementation
3. **What are the other stresses affecting the organization?**
   - ❑ How many number one priorities are currently competing for finite resources?
     - ❖ Has leadership assigned realistic and clear-cut priorities over time?
   - ❑ Where are current hot spots that are likely to flare if yet another change is deployed?
   - ❑ Where are the calm spots where a pilot deployment might demonstrate success?
   - ❑ Has leadership built commitment from within by differentiating the importance of this effort?
4. **What kind of resistance to the change required by the solution is likely to develop?**
   - ❑ What current mission activities are going on that might be disrupted by this change?
   - ❑ What current mission activities can be delayed or terminated to support this change?
   - ❑ How large is the gap between the current roles and responsibilities of key personnel and the new roles and responsibilities these personnel must adopt for the change to be successful?
   - ❑ How can these key personnel be supported to learn their new roles and responsibilities and develop predictable skill using the solution during the change process while they continue to perform other mission critical tasks?
   - ❑ What are the strategies for surfacing resistance to the change so the resistance can be dealt with?
     - ❖ Can actual resistance be differentiated from the process used to surface it?

- Are there milestones and checkpoints at which to assess overall progress?
  - Is there a strategy for overcoming the daily grind of complaints?
- Are desired changes motivated by making it more painful to remain in the *status quo*?
- Where are the sources of remaining uncertainty? Can the uncertainty be removed?

5. What is the availability and skill of qualified insiders to facilitate the change (change agents)?

- Have the people with proper skills and motivation been selected, not just ones from the pool of available people?
- Is their vision completely in alignment with that of the leader?
- Are the change agents producing actions that are clearly connected to reaching the goal, not just doing something?
- Are the change agents students of lessons previously learned, and motivated to avoid past failures?

6. How will the organization's culture hinder the change? (How do we adjust the change to address the realities of the way the organization does things?)

- How well does the solution align with current values, behavior, and unwritten rules?
  - Is survival dependent on our success (if not, why are we doing this)?
- If changing culture is required, the change must start with the leader and cascade down.
- Is resistance due to culture a predictable and well-known phenomenon?
- Is the leader providing the inspiring and energizing vision and leading the effort day in and day out to effect a culture change?

Enabling

Given deep insight into what needs to be changed and how such a change results in improvement, the enabling phase is used to design and plan for the improvement. Most modern techniques for implementing change are complex, and the benefits of their usage tend to occur in the project life cycle after the investment. (Sometimes the benefit occurs much later.) Without careful evaluation, tailoring, design, and planning, the business process change effort may provide little or no real benefit to the organization.

Technology evaluation

There are many alternatives from which to choose for most issues. Often, the best technology for an organization may not be the "best" when analyzed abstractly. Picking any aspect of a technology, finding the best with regard to that aspect, and then assuming the winner should be adopted is a mistake. (The highest quality product may be too expensive or too slow; the least expensive may not have adequate training and support services

available.) Picking the best process, method, or tool for an organization requires the careful balancing of a number of aspects and this task should not be taken lightly. The Software Technology Support Center at Hill Air Force base has performed a number of such evaluations for their customers, and documented the foundation for their approach [27].

The goal of the selection should be a technology that supports the new workflow the organization wishes to establish in a way that fits the culture of the organization. It is also critical to realize that buying new tools and just providing training has almost never resulted in an organization changing how it performs its work, at least not quickly.

Solution tailoring

Few technical solutions are effective for an organization right out of the box. While one expects the organization must change in order to receive benefit from a new process, method, or tool (the new workflow), the same is true of the technology. To obtain the most from a technology, some tailoring is almost always required [28]. Sometimes this tailoring is nothing more than the production of a collection of templates and usage guides. In other cases, it may be that more significant changes are required. The key is to remember the goal: improve how the organization performs its work. If changes to a tool's interface will speed the typical worker's use of the tool, such a change should be considered.

Usage design

Getting the value from a new process, method, or tool requires the people to use it effectively. Most modern technologies can be used more than one way. If there is no effort to design how the workflow is to be broken up into components that are to be assigned to different workers and managers, then the organization is leaving the design and implementation effort to each individual to dream up on their own. The likelihood that all of these people will pick a single, effective implementation is nearly zero. Before serious adoption planning can take place, the design of the technology's usage must be performed. One approach is to follow the object-oriented method of use case analysis as if the workflow were going to be automated [29].

The simplest usage design is the "plug-compatible" replacement–the technology replaces an existing solution with little dramatic change to the rest of the organization. At the other extreme is complete and radical change, such as complete reengineering of the organization.

Experience has shown that plug-compatible replacements can lead to incremental improvements in the organization, but they seldom lead to dramatic improvements. (How can an organization perform its mission dramatically better or faster if the majority of the organization doesn't change?) This does not mean a plug-compatible approach is bad; rather, it is important to set expectations and ensure that the cost/benefit ratio warrants the investment.

Experience has also shown that complete organizational reengineering is costly and often produces far less than might be expected [30]. If a technology demands a complete reorganization before it is possible to benefit from its adoption, other options should be

considered. The only reason to completely reengineer an organization is when the only other option is the failure of the organization to survive.

Business process change management planning

Planning and staging the adoption to maximize the benefit to the adopting groups at times that gracefully fit with other commitments and the availability of resources is critical. If business process changes are to result in dramatic improvement in how the organization performs its mission, one must assume that dramatic changes are required. This implies that a significant number of people in the organization must change the way they perform their jobs. Planning how to change these people is not trivial.

Knowing who must change, what change is required, and what will motivate them to change is fundamental to building an effective adoption plan. Each of the people who must change must also see how the change is connected to the mission of the organization in ways they can understand and value. If these people are not properly motivated to accept some near-term pain, it is common for them to expend more energy resisting the change than that required by the change. Some may consider using fear as a motivator. Experience has shown that professionals are not well motivated by fear. They respond much more favorably to seeing why their involvement is important and how the success of the effort is directly connected to the success of the organization as it performs its mission. Being part of new and critical work that is clearly valued by the most senior people in the organization is often all the motivation that is required.

Key factors in successful adoption of technology solutions are the quality of the Business Process Change Management Plan and the discipline the organization exhibits to honor the spirit of the plan. The wide variation in the nature of technologies being adopted and the organizations doing the adoption makes it impossible to recommend a single template for business process change management.

The Business Process Change Management Plan should not be viewed as a legal contract between two groups involved in the business process change(s). Rather, it should be viewed as a communications tool to assist all involved parties as they strive to understand what needs to be done, the rationale, and how it is to be accomplished. As a communications tool, a Business Process Change Management Plan must be written in a way to ensure that the readers understand the plan. Too often, plans make too many assumptions about what people know and what words mean. Effective plans are those that use truly shared mental models [31] and take the time to define terms when critical concepts are not truly shared.

The Business Process Change Management Plan must also not be viewed as a static document, for insights and technology seldom remain static for long. The right amount of detail to provide in an adoption plan can only be established by means of experience. With too much detail, one risks stifling alternative creative solutions and reducing willingness to alter the plan due to the size of the investment to create it. If there is too little detail, those charged with the implementation may not fully appreciate the intent of the plan, the work others are to perform, and how it all is supposed to fit together into a coherent whole. The

plan is designed to be an ongoing guide to allocate resources. This plan can be used to implement relatively simple changes by completing only those planning components that reflect the key barriers identified in the Assessment analysis. The document also can be used to manage very complex efforts by providing a detailed and structured planning process.

An effective Business Process Change Management Plan considers all of the issues described above, assessing the risks of ignoring these things, and weighs these risks against the costs and potential benefits of addressing and resolving them. If the adoption of the technology were treated as a mission-critical project and lessons from each effort were captured and fed forward to future efforts, the organization would discover that adoption does not have to be unmanageable or unpredictable.

The primary benefit of the Business Process Change Management Plan is realized only if the plan is actually used and the organization is prepared to make the near-term sacrifices to honor it. If the adoption is not mission-critical to the point that management is able to remain focused on its implementation, staff it with the proper people, support it with adequate resources, and assist it with creative and proactive problem resolution efforts, there is little hope that adoption will be either rapid or effective.

Acting

Too often, people assert that strategic planning doesn't work because there is no improvement following the planning. On closer examination, their failures are really failures to act. If there is no serious and concerted effort to implement the plan, there is no chance for an improvement to occur. While planning is important, it is meaningless without dedicated implementation.

Business Process Change Management Implementation

Setting the stage for effective implementation of the adoption plan begins with the sponsor. The launch of the implementation establishes the entire tone for the effort, and getting the right people to say and do the right things is key.

One very useful preparatory activity is to examine how the organization has launched similar efforts in the past and leverage the lessons from those experiences. If this launch is just like all of the other improvement efforts that have failed miserably, why should anyone get enthusiastic about getting involved? Insanity has been described as doing something the same way as it has been done before and expecting a different result. How will this effort be different and why?

Treating the adoption as a mission-critical project has been very effective. This requires the project to have a budget, a plan, a schedule, dedicated resources, and regular management oversight. It has been said that improvement efforts must operate at a maturity level at least as high as what the improvement is trying to accomplish. If the technology requires process discipline and constancy of purpose in order to yield value to the organization, then the effort to adopt it must employ these same behaviors.

Skill development

Motivating people without providing them the skills they require to perform their new tasks can lead to real problems. Armed with new insights about the roles, responsibilities, and workflow, the organization must help its people develop the skills they will require to be successful. Most people understand the need to "train" the workers in the use of a new tool. This usually translates into some form of class. What is often not appreciated is how much real world practice is required before one becomes skilled in the use of a new tool. It is also common for people to ignore the fact that many skills are perishable. Without regular refreshment, skills fade and disappear.

As mentioned earlier, it is critical to recognize all of the various groups in the organization that must change and to question how these groups will become skilled in performing their new roles. For example:

- How will project managers learn to change how they plan projects and estimate the skills and resources required by a project due to the changes a new technology makes in the workflow?
- How will managers track the progress of projects and determine whether adequate progress is being made?
- In case there is a problem, how does one change things without damaging work already performed?
- How do executives assess the return on the investment in a new tool and justify the continued training, sustainment, and overhead costs?

If these questions are not explicitly answered in a uniform way and the affected personnel do not know or understand the answers, there is considerable risk the adoption will fail. (Many technically successful adoptions are cancelled because management could not appreciate the benefits in terms they valued. This management skill is just as critical as the operator knowing how to use the tool properly.)

Generic classes tend to provide an education, but do little to help the students develop insights about how the tool should be used in this organization or develop the skills they will need as they try to use the tool on the job. There are two rules of thumb:

1. The participants will be more able to translate their experiences to their job when the training is customized to fit the workflow, culture, and vocabulary of the organization and the training employs hands-on practice.
2. The longer the time between the training activities and the time when the student is called on to perform, the less likely the first usage will be successful.

If the first usage is to be successful and it is truly critical, it should be over-staffed in an effort to minimize the workload placed on each member of the team. It is also helpful to obtain the services of people skilled in the use of the tool and have them play the role of mentor or coach. (Nothing is more comforting than knowing that there is someone there to help

when reality does not seem to duplicate what was taught in the classroom or experienced during practice.)

Pilot testing

Pilot testing can be used to reduce organizational risk in the adoption of a new technology. The pilot tests are used to perform two critical tests at the same time:

1. Determine if the organization can obtain the promised benefits from the technology.
2. Evaluate the adoption approach and materials on a limited scale before taking the adoption to the entire organization.

Many consider the first test (Can the technology produce the desired benefit?) to be the primary role of pilot testing. In reality, the costs of pilot test and the impact on the organization make other methods for evaluating the benefit of the technology far more attractive. (For example, visits to other real customers of the commercial components would show the benefits at a far lower cost to the organization.) The people running the pilot test should have already determined that the technology solution could provide value; the only real questions are how hard will it be for this organization to adopt it and benefit from the adoption. Be careful. Most pilot efforts are not instrumented well enough or properly baselined to prove anything about the value of the technology solution in the organization. This should be the major focus of any pilot effort.

The best use for pilot tests is to evaluate the adoption of the technology solution and to showcase the organization's commitment to the changes the technology implies. When senior management embrace the new technology solution, change their behavior in a visible way to support the solution, and are regularly seen to assist others who are involved with the pilot test, the subsequent roll-out of the technology solution will be easier for the rest of the organization. If these leaders are seen as having taken a wait and see attitude, not being intimately involved, and not willing to take risks to make the pilot test work, the subsequent roll-out is likely to be long and painful.

Selecting the right project for pilot testing is important. The project must be early enough in its life cycle for the team members to be able to develop the skills they will need before being called on to use them. Some things to avoid include

- retro-fitting parts of a project in order to use it as a pilot test. (Most people will view these efforts as a waste of time and effort.)
- picking very short projects, as they are usually not significant enough to truly demonstrate the use and benefit of the technology
- picking a project that is too long, as the results may not be made visible in a timely manner
- using a project that is in trouble. (The extra effort associated with the pilot test will stress even relatively low-risk projects, and new technologies seldom make up for ill-considered projects, regardless of how effective the technology is at doing what it does.)

Some of the most significant benefits of a pilot test are found in determining how to enhance and refine the adoption method based on experience with real people from the organization. It is to be hoped that the pilot test would help answer the following questions:

- How useful were the skill development activities?
- Could the skill development exercises have been made more realistic?
- Were there vocabulary problems that weren't properly recognized?
- Should the materials be rewritten using nomenclature and examples from the organization?

While these lessons are usually too late to help those doing the pilot test, they can influence what is delivered to the bulk of the organization during the roll-out and can significantly reduce the total cost of adoption.

Organizational roll-out

How the organization moves to introduce the technology solution to the bulk of the organization can dramatically influence the likelihood of success. Force-fitting technology solutions where they don't fit can be disastrous. There are many questions to consider, for example:

- Should the entire organization adopt the technology, or is it really only relevant to part of the organization?
- How long will the technology be used before it is replaced by something different?
- Which groups are at the appropriate places in their project life cycles for the adoption to be reasonable, and how long will it be for those who are not yet at reasonable points in their cycle?

A common strategy for organizational roll-out is the mandate. While few mandates have been successful, their failures tend to be due to an unwillingness of the senior leadership to do the things required to make the mandate successful. By means of clear and consistent leadership, personal adoption by the most senior leaders, consistent reward for those who adopt and sanctions for those who do not, mandated change is possible. When executives are unwilling to implement all of these aspects, the likelihood of success drops.

At the other extreme is the optional adoption. These adoptions can only work if there is a clear motivation for projects to risk the adoption effort. If the organizational leadership is willing to fund the adoption activities, willing to provide additional technical and managerial resources, and willing to ease external pressures in order to support those who elect to adopt, optional technology adoptions can be successful. Senior leadership must establish clear motives to support the adoption, honor those who succeed with the adoption, and withhold sanctions from those who honestly tried but failed.

Learning

The only hope for long-term survival is to establish an organizational culture that values and leverages the lessons others have learned. In his Turing Award lecture [32], Dr. Hamming stated:

> "Indeed, one of my major complaints about the computer field is that whereas Newton could say, "If I have seen a little farther than others it is because I have stood on the shoulders of giants," I am forced to say, "Today we stand on each other's feet." Perhaps the central problem we face in all of computer science is how we are to get to the situation where we build on top of the work of others rather than redoing so much of it in a trivially different way. Science is supposed to be cumulative, not almost endless duplication of the same kind of things."

Peter Senge in his book *The Fifth Discipline* [31] echoes the thought as he describes the importance of creating a learning organization, and the special kind of teaming that develops. Modeling best practice is an obvious lever of past lessons. Understanding the standard failure modes is another. The wise leader is one that knows both, for there are surprises hiding in every human endeavor.

Lessons learned

The rapidly changing world will force successful organizations to establish technology adoption as an area of core competence. Just as soon as one solution adoption is completed, management should be considering what the next adoption should be and when it should take place. If the lessons from previous adoptions are not fed forward to the benefit of the subsequent adoptions, the organization is bound to suffer needlessly.

The most powerful tools in technology adoption is the use of experienced people and the use of lessons learned from previous adoption efforts by the organization. If each adoption effort is performed in a cocoon of ignorance, the team is doomed to repeat previous failures. The organization should record and maintain information that helps answer the following kinds of questions:

- What did we do last time?
- What worked and why?
- What didn't work and why?
- What are we going to do this time and why do we believe these changes will resolve the issues we failed to resolve well before?
- How should we capture and leverage this knowledge so technology adoption can become an area of core competency here?

Usage Considerations

Adopting technology solutions invariably comes into conflict with day-to-day operational efforts. Most adoptions fail due to non-technical reasons (such as political pressure to throw resources at some tactical effort that is in trouble). The complexity of these adoptions, their duration, and their highly visible nature require extraordinary efforts. Using experienced professionals, dedicating people to the tasks, and using influential outsiders as coaches to keep the adoption moving down the planned path, has been useful in many efforts. Technology adoption can be mastered if we are willing to stand on the shoulders of giants as opposed to insisting that this situation is unique and the past has nothing to tell us.

Business Process Change Management Plan Artifact

Description

The Business Process Change Management Plan provides a planning structure to increase the likelihood of successful end-user business process change implementation. The plan is designed as an ongoing guide to allocate resources. The plan can be used to implement relatively simple changes by completing only those planning components that reflect the key barriers identified in the organizational assessment. The artifact can also be used to manage very complex efforts by providing a detailed and structured planning process.

Tips

Some of the components of this artifact (e.g., Diagnostic Activities) must be done before other sections can be completed. Other sections (e.g., Key Role Mapping) will need to be updated as new information becomes available.

Business Process Change Management Plan review checklist:

1. What advantages and benefits might there be to accepting this plan?
2. Who are those people who will gain from this plan? Who will lose?
3. Who must his plan be shared with?
4. Is the plan easy to understand? Is it translatable into needed organizational elements?
5. How can this plan be labeled, preserved, or packaged to increase the likelihood of acceptance?
6. How can this plan be shortened and summarized to help communicate it effectively to others?
7. Is this plan reversible? Is it possible to stop the process once we start it?
8. If necessary, could a piece of the plan be implemented without committing the organization?
9. How can alterations/suggestions to improve the plan be incorporated to modify the implementation and enhance the likelihood of success?
10. What are the feedback loops to sponsors, agents, and targets within the plan? How often will feedback to each key role occur?

Template

1. Solution Identification
   1.1. Name
   1.2. Version Number
2. Assessment analysis
   2.1. Complete implementation risk analysis.

   [Identify key barriers and key leverage points and assign risk score in implementation history, sponsorship, target resistance, culture, and change agent.]

   2.2. Identify recurring patterns in the barriers and key leverage points for each of the five factors.

   1.1.2. Examine the barriers identified for each of the five factors.

   [Determine whether any common barriers to change cut across the profile. What recurring barriers do you see in the pattern? These common patterns provide the specifics of critical actions needed to increase the probability of success. They represent strategic barriers in the implementation, and resources should be allocated to eliminate or minimize their impact.]

   1.1.3. Examine the key leverage points for each of the five assessment tools.

   [What recurring patterns exist in the driving forces for successful implementation? How will these be leveraged/ reinforced to contribute to success of the effort?]

   2.3. Set priorities.

   [Rank the identified barriers in order of importance. This list represents the priority of resource allocation needed to increase the probability of success.]

   2.4. Show sequence of action.

   [Translate the priorities into a sequence of action steps. There may not be a simple correspondence of priority to action, as some actions may need to be taken before work can begin on the first priority.]

   2.5. Show how the analysis will be used.

   [The analysis has produced four strategic orientations to have an impact on the change effort:
   - *A list of strategic barriers that must be addressed*
   - *A list of key leverage points to reinforce*
   - *The priority resource allocation for the key barriers*
   - *A sequence to maximize the impact on those priorities]*
3. Preliminary planning
   3.1. Does a formal written definition of this change and its outcome measures exist?
   3.2. Review the Project overview with the implementation team.

[How has the definition of this change effort altered as a result of the assessment analysis' overall forecast of risk?]

3.3. Does this definition of the change need to be reconfirmed with the authorizing sponsor?

[Do the authorizing sponsor and agents share a common definition? What format will this interchange take and when will it occur?]

3.4. Does a common definition of the present state exist in the organization?

[Who has contributed to this? How has it been confirmed in the organization?]

3.5. Is there a clear and common understanding for the need for this change?

4. Diagnostic activities: confirming perceptions (ref: assessment analysis)

[This initial prediction of the likelihood of success has probably been done in the absence of comprehensive diagnostic data. It may have been completed by an individual or team and based only on their collective perceptions. To provide a complete and thorough picture, it is necessary to validate your perceptions with diagnostic activities throughout the organization. Sponsors and agents who simply assume they understand the frames of reference of the target groups are living very dangerously.]

4.1. Identify a set of diagnostic activities to generate more accurate and comprehensive data on the risks of implementation, key barriers, and key leverage points.

5. Key role mapping (ref: sponsorship assessment, key roles)

[Using an organization chart as a map, identify the key roles in the path that this implementation plan must follow from authorizing the sponsor to the ultimate targets of the change. The key role map should be used as an ongoing reference to guide progress. For example, it may be too early to identify all the change agents at the local level. Identifying subsequent agents would be added to the Master Plan task lists. However, identify all the current key players and update this list as planning the implementation proceeds. Identify

- *authorizing sponsor*
- *all reinforcing sponsors*
- *agents*
- *target groups directly affected by the change*
- *target groups indirectly affected*
- *individuals with overlapping or multiple key roles (principle: if roles overlap, assume target first)*
- *linear, staff, and matrix agent configuration]*

6. Sponsorship (ref: sponsorship assessment, key role map)

6.1. Who is the authorizing sponsor for this change?

[What is his/her current level of dissatisfaction with the present state?]

6.2. What data have been surfaced to the authorizing sponsor to create this need/discomfort?

6.3. Currently, what resources have been committed/allocated to support this implementation?

6.4. How and when will sponsorship be cascaded down to the first level of reinforcing sponsors?

6.5. Should a sponsorship development summary be completed to improve overall sponsorship?

7. Change agent development (ref: change agent assessment)

Who are the primary change agents for this change–people who have initially been charged with the ultimate responsibility for implementation success or failure?

7.1. Identify the other change agents who will be assisting implementation at the local level.

7.2. What strengths or assets do the agents possess that could be effectively used in this implementation?

7.3. What liabilities or weaknesses do the agents have that could increase risk of failure?

7.4. Complete the change agent development summary.

Plan response to the liability–who, when, resources, task number

8. Communication plan (ref: transition management tactics)

8.1. How will you explain the rationale behind the change?

8.2. How will the cost of not changing be emphasized?

8.3. How will the change be explained in a manner that emphasizes external drivers, not internal deficiencies of the targets (does not invalidate targets' past history)?

8.4. How will this implementation be communicated in the FOR of the target groups?

8.5. What varied mechanisms (speeches, memos, videotape interview, etc.) will be used to deliver both the objectives and rationale of this change?

8.6. How often will you provide formal communication during implementation?

How will you ensure that other events will not preempt regular communications?

8.7. What specific things will not change? How will these be communicated?

8.8. What mechanisms will be used to ensure that communication is two-way and open during implementation? What feedback loops will be established? How often will feedback be provided?

8.9. How will progress be communicated to
- sponsors
- agents
- targets

8.10. How will you communicate that the sponsors understand the price the targets have paid for this implementation?

9. Reinforcement plan (ref: transition management tactics)

   9.1. How has reinforcement been changed to support implementation (rewards, punishment, effort) by
   - sponsors
   - agents
   - targets

   9.2. How has performance in this implementation been incorporated into performance objectives for
   - sponsors
   - agents
   - targets

   9.3. How has performance in this implementation been incorporated into the performance appraisal process for
   - sponsors
   - agents
   - targets

   9.4. How will the organization reinforce that it is safe and appropriate to resist overtly, not covertly?

   9.5. How will targets be involved in implementing the change?

   9.6. What specific symbolic action will sponsors use to demonstrate commitment?

   9.7. How will major and minor victories be identified and communicated?

   9.8. How will champions be rewarded?

10. Target resistance plan (ref: target resistance assessment)
   - target groups that will be affected by the change
   - type of impact: direct/indirect
   - level of disruption form the change (high, medium, low)

   10.1. Perception of change (positive, negative)

   10.2. Target groups with high levels of disruption will demonstrate the most resistance to the change and will require a high level of management attention. Which groups have the highest levels of disruption?

   10.3. Different strategies for managing resistance will be required depending on targets' initial perception of the change as either positive or negative.

   10.4. Complete target resistance strategies (target, source of resistance, response, who, when, resources, task#).

11. Cultural resistance plan (ref. implementation history assessment, culture assessment

11.1. Historically, what critical barriers to successful implementation have existed in the organization? What specific strategies will be used to ensure that these barriers do not repeat?

11.2. Based on the cultural assessment, which dimensions of the culture will have to be significantly altered to ensure successful change? What strategies and tactics will be employed to minimize cultural resistance from that dimension?

11.3. What reinforcements will be used to motivate a shift in the culture to support implementation?

12. Monitoring and tracking plan (ref: project overview)

12.1. What are the outcome measures for this effort? How will we know when we get there (both technical and human objectives, time, and budget)?

12.2. What intermediate milestones will be utilized to demonstrate progress? How often will progress be monitored?

12.3. What resources and sponsor commitment will be required to provide sufficient tracking and monitoring?

12.4. How will progress be communicated and feedback provided to
- sponsors
- agents
- targets

13. Integration and master planning

13.1. Review the analysis of data. Have the identified, strategic weaknesses been addressed and reflected in the planning?

13.2. Has the established priority and sequence been reflected in the ordering of tasks and the allocation of resources?

13.3. Review the key roles map. Has all information been completed and roles identified?

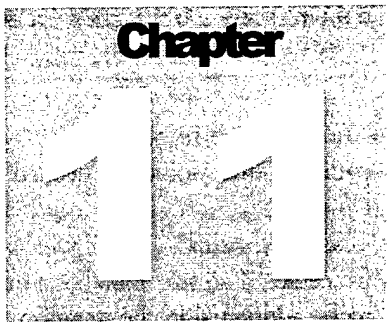
13.4. Most successful implementation projects are characterized by a centrally coordinated and integrated approach to planning and execution. In this initial planning process we have developed eight sets of action plans:
- Diagnostic activities summary
- Sponsor development summary
- Change agent development summary
- Communication plan
- Reinforcement plan
- Target resistance strategies
- Cultural resistance strategies
- Monitoring summary

13.5. Each item on these action plans represents a task to be accomplished. As such, all tasks should be assigned and placed on an integrated task list. These tasks (cross referenced to the corresponding action plan) should be ordered based on

- priorities established
- sequence of actions
- key role map
- a logical progression from
    1. preliminary planning
    2. diagnostic activities
    3. implementation strategies
    4. monitoring

13.6. Each action plan has detailed "resources required." Each resource must be secured form sponsors of the change. Comprehensive planning would include a resource summary cross-referenced against the task sheet and the individual action plans.

13.7. Each action, strategy, and resource should be coordinated in an overall master plan for the time frame, including specific responsibilities and time frames.



Executable Representation Guidelines

Background

It is essential that each iteration builds something that is concrete and executable that end users, engineers, managers, and other affected stakeholders can "touch and feel." An executable representation aids in developing a shared vision of the proposed solution and provides a means for understanding the complex component and end-user business process interactions within the solution. It orients the project toward identification and resolution of critical issues and risks rather than the production of "paper documents." The executable representation stimulates early discovery of mismatches and negotiations to resolve the mismatches within the operational context of the system (e.g., use cases). It also objectively demonstrates the current state of the project. This acts as a forcing function that drives the project to gain closure at regular intervals. Development of the executable representation also fosters early elimination of architectural defects.

Within the context of the objectives for the iteration, the executable representation provides a vehicle for

- experimenting to clarify information concerning both the problem and the solution
- demonstrating consensus among stakeholders on the solution by providing a visual and tactile representation of the negotiated solution (i.e., seeing and feeling how it works)
- building stakeholder buy-in for the solution by showing something concrete to users, customers, and managers
- assembling and fitting components within a design
- uncovering key technical and/or performance issues associated with the solution
- demonstrating progress by achieving a desirable level of functionality

An executable representation is a working model of the solution that emphasizes specific parts of the solution needed to meet iteration objectives. It contains subsystems, components, integration mechanisms (such as tailoring and wrappers), and links to key external systems – all of which interact with each other and with the end-user business processes. The size and scope of an executable representation depends on the specific risks to be addressed in the iteration. The fidelity of the model to the actual solution increases across the phases. Initially, it may be a very loose outline or skeleton of the solution to demonstrate proof of concept; in later phases it becomes a potentially releasable version of the solution.

Executable representations that look and feel like the potential system allow the users to evaluate potential solutions, which will usually result both in changes to their requirements and in better solutions. These changes will be validated in later iterations. Executable representations increase user involvement and can enhance user/developer interaction. They are particularly valuable when users are not familiar with the technology that is likely to be used to implement their system. They are also useful in analyzing the capability and acceptability of COTS system components, reducing the risk in component selection. They are able to experience firsthand the impact of the new technology or component on their business processes.

Form

The executable representation will model the solution with increasing fidelity through successive iterations. In early iterations, the executable representation will consist of a rough outline or skeleton of the system. In later iterations, the skeleton will be populated with selected components and linkage mechanisms that demonstrate critical use cases and the mitigation of key risks. The nature and goal of each executable representation must be explicit, particularly in the inception and elaboration phases, where the focus is on discovery, experimentation, and risk assessment. Otherwise, it is easy for users and customers to misinterpret what they see as an implementation that is complete or robust enough to be fielded. In the construction and transition phases, the focus shifts to completeness, consistency, and usability of the solution. In these later phases, the end users may want to continue development of the solution rather than accept a formal release of the solution. The different uses of the executable representation lead to a shift between phases in the character of executable representation:

- Inception Phase: proofs-of-concept
- Elaboration Phase: architectural prototypes
- Construction Phase: production quality releases (including alpha, beta, and other test releases)
- Transition Phase: general availability releases, bug fixes, patches, minor enhancement releases

for the Inception Phase...

The focus of this phase is conceptualization; therefore, the executable representations emphasize proof of concept. Most often, proofs of concept take the form of a quick-and-dirty executable. Such executables are by their very nature incomplete and only marginally engineered. It is far better to discover during proof of concept that assumptions of functionality, performance, size, or complexity were wrong, rather than later, where abandoning the current development path would prove to be financially or socially disastrous. In addition, by keeping around the interesting (although perhaps rejected) executables, the organization maintains a corporate memory of its original visions, and thus preserves the assumptions that were made when applications were first conceived.

It must be emphasized that all such executables must not be allowed to directly evolve into the production systems, unless there is a strongly compelling reason. The executable representations in this phase should focus on demonstrating feasibility of one or more candidate solutions. Efforts to build any part of a solution should be deferred to the elaboration phase.

A proof-of-concept executable representation is developed to

- determine feasibility of a candidate solution
- gain an initial understanding of applicable commercial components
- assess the technical risks associated with each candidate solution
- refine the architecturally significant requirements of the solution
- refine the critical, "must have" end user requirements
- identify the significant changes to the end user's business process inherent in the candidate solution

The executable proof-of-concept integrates the foundation components of a candidate architecture and provides an executable framework for elaborating the critical use cases. It contains potential frameworks, architectural or design patterns, representative COTS components under consideration (although only small parts of the products may be used), very skeletal custom components, and early component integration mechanisms.

for the Elaboration Phase...

Designing the architecture, validating it, and then baselining it are the number one objectives of the elaboration phase. To validate the architecture and to assess its qualities in terms of feasibility, performance, flexibility, and robustness, we must build it. Therefore, in addition to a software architecture description, the most important artifact associated with the architecture is an architectural prototype that implements the most important design decisions sufficiently to validate them–that is, to test and measure them.

The architectural prototype is a partial implementation of a solution, built to demonstrate selected system functions and properties. The extent of the architectural prototype depends on the scope, size, risk, and novelty of the solution. It should at least address the critical use cases and the non-functional requirements (also referred to as quality attributes) that have an

impact on the solution architecture. It is built to mitigate risks related to business processes, performance, throughput, capacity, reliability, and other 'ilities'. The architectural prototype is also used to assess whether the interfaces and collaborations among components (COTS components as well as custom) are consistent and complete within the context of the system's critical and significant requirements and scenarios. There is typically not much implementation depth for any custom components.

This prototype is not a quick-and-dirty throwaway prototype, but it will evolve through the construction phase to become the final solution. The architectural prototype is built with the intention of retaining what is found to work (and satisfies requirements) and making it part of the fielded system. At a minimum, the architectural prototypes should demonstrate

- the critical business use cases
- the initialization of the system
- a scenario to drive the worst-case data processing flow through the solution
- a scenario to drive the worst-case control flow through the solution

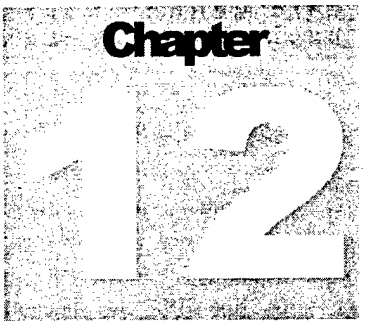
for the Construction Phase...

The primary product is a stream of executable releases representing successive refinements to the architectural prototype. A release is a stable, executable version of the solution with production quality and rigor. A release is not necessarily a complete product, but can be just one step along the way, with its usefulness measured only from an engineering perspective. Early in the development process, major executable releases are used to begin to test the release against the completeness, correctness, and robustness of the release. This early data gathering aids in identifying problems of quality.

Later in the development process, executable releases are turned over to select end users (the alpha and beta customers) in a controlled manner. By controlled, we mean that the development team carefully sets expectations for each release, and identifies aspects that it wishes to have evaluated. Alpha releases generally include executable capability for all critical use cases. A beta release is typically a fully operational system (although it does not necessarily have to have the complete range of functions envisaged in the production system) developed for the purpose of gauging system performance.

for the Transition Phase...

An executable representation is a stream of executable releases that fix bugs, implement patches, or incorporate minor enhancements. A release is a stable, executable version of the solution with production quality and rigor that is fielded to the general user community.



Vendor/Supplier Relationship Plan[18] Guidelines

A vendor/supplier relationship is a cooperative exchange to explore current and future vendor/supplier and acquirer plans. Vendor/supplier relationships provide insights into current and future component releases and are a means for the acquirer to influence the vendor/supplier's plans and component directions. It is the primary means of partnering with vendors/suppliers whose components are important to the system.

Vendor/supplier relationships encompass the set of activities to determine candidate vendors/suppliers with whom to develop sound relationships (i.e., those vendors/suppliers with whom a sound partnering relationship is most important) and the nature of those relationships. This may involve meeting with vendors/suppliers and participating in user or vendor-related groups.

A *supplier* relationship is a partnership among entities in which one entity acts as a supplier to others. Such a relationship may encompass marketing, exchange of funds, formulation of agreements, and long-term component support. The acquiring organization needs to understand its limited ability to control the current and future capabilities of the component supplied and develop ways in which to influence the suppliers.

A *vendor* is a special case of a supplier where the components are sold or leased to a broad segment of the marketplace. Vendor is not a new term for contractor. Contractors can be directed to perform agreed-upon work within cost, schedule, and quality parameters. Vendors do not work this way. Thus, it is important for the procuring organization to understand its limited ability to control the marketplace and to develop ways in which to influence it. The license agreement is the main vehicle for establishing the foundations of the relationship with vendors.

Developing and managing relationships with vendors/suppliers is a new activity for many projects.

[18] Adapted from [6].

Activities

- Understand and monitor the vendor's/supplier's long-term approach and plans for maintenance and support.

  - For EPIC, this activity begins in the Inception Phase as market research is conducted and general trends within the relevant market sector are understood. The information is amplified for each component and its vendor as components are identified to be included in candidate solutions. In the Elaboration Phase the Component Dossier is updated to reflect those components that are included in the selected solution. This information is updated routinely through the Construction and Transition Phases to help proactively plan component updates throughout the life of the component in the system.

- Develop a strategy to create and manage vendor/supplier relationships. Record the rationale for selecting candidate vendors/suppliers (e.g., why certain vendors/suppliers are more important than others) and the nature of all vendor/supplier relationships (e.g., the depth of the relationship to be pursued).

  - With EPIC, this strategy is developed and captured in the Vendor/Supplier Relationship Plan artifact in the Inception Phase as part of each candidate solution. The vendors/suppliers selected for inclusion are those whose components play a significant role in the solution. The nature of the relationship will depend on the leverage the project has or needs to have over component support and upgrades, and will evolve through the later phases. The plan is updated in the Elaboration Phase as components are selected and maintained in the Construction and Transition Phases as changes are required.

- Engage in meetings and exchanges with the vendor/supplier and vendor-supplier-related groups. Do this for all vendors/suppliers with whom the project has a relationship.

  - In EPIC, begin interacting with vendors/suppliers in the Inception Phase. The relationship with each vendor/supplier becomes more directed and involved when the component is selected in the Elaboration Phase and then continues through the Construction and Transition Phases as long as the component is part of the solution.

- Establish liaisons with other customers (or potential customers) of the vendor/supplier. Do this for all vendors/suppliers with whom you have a relationship.

  - With EPIC it is just as important to stay connected with other customers who use the component as it is to interact directly with the vendor/supplier. Lessons in the use of the component can and should be shared. In addition, it is important to understand the forces that are driving the vendor/supplier to make component changes. Unified customers seeking a particular enhancement are a formidable incentive for the vendor/supplier. The relationship with other customers begins in the Inception Phase and continues through the Transition Phase.

- Coordinate government vendor/supplier relationships with the contractor vendor/supplier relationships in cases where both exist.

- ❖ For EPIC, it is important that both the acquiring organization and the integration or engineering contractor have relationships with the vendor/supplier. These relationships need to be coordinated early in the Inception Phase and should be reviewed to make sure they stay linked through the Transition Phase.

- ❑ Encourage and facilitate working relationships among the vendors/suppliers.

  - ❖ In EPIC, these relationships begin in the Elaboration Phase as components are selected. Close relationships may be critical to success in the Construction Phase as components are integrated into higher-level components or into the broader organization's architecture. Where possible, these relationships should be continued in the Transition Phase to help resolve component mismatches that occur as part of new component upgrades.

Tips

Influencing vendors/suppliers

A single customer does not control vendors/suppliers. It may be possible to influence component directions or gain insights into future directions through a sound partnering approach to each vendor/supplier.

Relationship development

Developing a relationship with key vendors/suppliers should begin as early as possible–as early as concept development or stakeholder needs formulation. Know the vendors/suppliers in the segment of the marketplace that is most important to the project.

- ❖ The Virginia Class submarine program engaged potential vendors in a series of critical item tests well before the request for proposal (RFP) was released. In addition to demonstrating the vendors' capabilities and revealing potential integration problems, these tests also signaled the start of the project's cooperative relationship with the vendors.

Investing in relationships

Vendor/supplier relationships are not free; both the project and the vendor/supplier expend time and resources to establish and maintain the relationship. Time will be required to cultivate and maintain the relationship; trust must be built on both sides. Not all relationships are worth the same investment. Carefully choose the vendors/suppliers the project has relationships with and how much effort is put into them. Base this choice on the importance of the vendor/supplier and its component to the project and the risks of not paying close enough attention.

User group participation

The project should participate in component user groups–and may consider making presentations–to help sustain the project's relationship with its vendors/suppliers.

Factors to consider

Vendor/supplier relationships are not only a concern for the project's integration contractor. The acquiring office may want to establish its own relationships with key vendors/suppliers. Be alert to the fact that the contractor may be sensitive about customer relationships with vendors/suppliers; be sure to include the acquirer's right to deal with vendors and suppliers directly in the initial contract. Similarly, be careful when there are direct end-user relationships with the same vendor/supplier.

You should also know if the integration contractor has any pre-existing strategic relationships with particular vendors/suppliers. These are not necessarily bad, but the acquiring office should be aware of them in case they unduly or inappropriately influence decisions that the contractor is making.

Keep perspective

Keep perspective on any vendor/supplier relationships. Participating in a relationship with a vendor/supplier may create a tendency to lock the acquiring office or contractor into a particular technology or components that may or may not be the best technical solution.

Relationship changes

Relationships can change over time; do not assume that once a relationship with a vendor/supplier is established that there is no longer a need to maintain the relationship.

- For example, one DoD program faced the loss of its negotiated vendor/supplier relationships when its projected purchases fell to 20% of original estimates, which was too low to be of sufficient interest to the vendors/suppliers any longer.

Required skills

Special skills are required to be successful at cultivating vendor/supplier relationships. Chief among these are communication and people skills. Project managers, deputy project managers, chief engineers, and architects must have these skills.

Mandatory or voluntary

Some supplier relationships may be mandatory, others voluntary.

- For example, the Defense Information Systems Agency is the provider of the Defense Information Infrastructure Common Operating Environment (DII COE), which is mandated for a large number of DoD systems.
- In the case of the Intelligence and Electronics Warfare Common Sensor (IEWCS), an Army program, the Marines voluntarily made substantial use of the Army's component–it became an important business relationship for both.

Contingency plans

The project may be faced with issues when the project's use of a component diverges from the abilities of the vendor/supplier, as may occur with shifts in component thrusts and supplier demise. Make contingency plans for these types of eventualities.

Version release

What a vendor/supplier can release at any point in time is unlikely to include the latest versions of all included components. There is a necessary delay to bring in the component, reintegrate and test, and then re-field.

Cycle frequency

The release cycles by suppliers may not be as frequent as those of COTS component vendors.

Budget cycle support

An acquirer may need to support a supplier's budget cycle, as the continued funding of improvements to the component is of interest to both.

- For example, when non-Army services made use of IEWCS, they sometimes needed to help defend the budget requests for this Army program.

Supporting completion

An acquirer may want or need to contribute funding to a supplier's completion of critical components. Such funding may afford the project some leverage from this engineering investment across programs, and it may be necessary to ensure timely completion of critical components.

Vendor/Supplier Relationship Plan Artifact

Description

[Insert a summary of what this artifact does.]

Template

1.0 Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

2.0 Components

[List components, annotated to show vendors and suppliers (including third party distributors that provide maintenance or other support services).]

3.0 Vendor/Supplier Name

[Include one section for each vendor and supplier.]

3.1 Component criticality

[Assess the criticality of each component to the system.]

3.2 Relationship objective

[List objectives for the relationship with each vendor or supplier.

Describe the extent to which the project needs to be aware of or influence vendor or supplier mid- and long-term component directions and trends in frequency of release, etc.]

3.3 Relationship strategy

[Describe the project's strategy for meeting the objectives described above, including the level of involvement and associated need for funding and other project resources. For example:

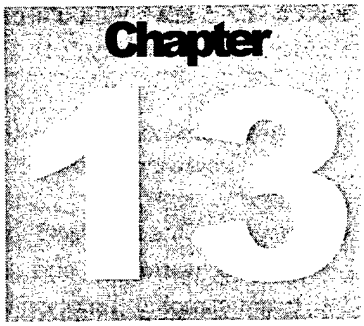
- *Meet regularly with the vendor or supplier. (Include schedule and level of participation within project and counterpart.)*
- *Participate in standards bodies. (Describe which bodies as well as the level of participation in each.)*
- *Establish direct relationships with other customers.*
- *Participate in user groups.*
- *Facilitate interchanges between vendors and/or suppliers.]*

3.4 Project information use

[Describe how the project will use information from the vendor/ supplier (include identification of roles and responsibilities for key players).]

3.5 Associated relationships

[Describe the integration or engineering contractor's relationship with the vendor/ supplier. Include mechanisms for coordinating with contractor and project activities.]



License Agreement Guidelines[19]

License negotiation is the set of activities that determine what the vendor offers with respect to terms, conditions, and costs for a given component for use by the organization over a particular period of time. Based on the situation and needs of the organization, this set of activities is used to negotiate the license(s) that is best for both parties.

It is important to realize that licenses can be negotiated. Typically, the vendor has a set of usual licenses that it offers, but other ideas or additions may be negotiated, as will be seen in some of the following tips.

Activities

- Conduct a preliminary investigation of licensing alternatives and costs in support of the business, understanding the range and costs of licensing options available (or that can be negotiated).
  - For EPIC, begin this activity as part of the initial market research conducted in the Inception Phase. Understanding of licensing options that other customers in the market segment expect from the broad base of vendors should be captured in the Market Segment Information artifact. Narrow the investigation of options as part of component evaluation as the project begins to converge on a more narrow set of COTS components. When the components are actually selected in the Elaboration Phase, the specific component's license information will be captured in the Component Dossier artifact.
- Secure a budget that will be appropriate for licensing activities and the cost of licenses.
  - In EPIC, this information should be captured in the Development Plan artifact developed before or very early in the Inception Phase. Because there will be different components evaluated as part of each candidate solution, there will be different license costs and the cost

[19] Adapted from [6].

implications for each solution will have to be carried until a solution is selected in the Elaboration Phase. All costs should be updated as information is updated and refined.

- Negotiate the license(s). License negotiation can include obtaining appropriate types of licenses, warranties, and data rights; managing licenses; and negotiating other kinds of maintenance and support critical to the component.

  - With EPIC, license negotiation should begin with component selection in the Elaboration Phase when development licenses will be required. It may be in the best interest of the project to negotiate terms for runtime licenses at this time too, depending on the risk of finding problems in the Construction or Transition Phase that could cause the selected component to be replaced. Runtime licenses will have to be negotiated for the initial rollout, or beta version, during the Construction Phase to support the Transition Phase and will have to be complete before full rollout can occur in the Transition Phase. Licenses should be configuration controlled as part of the configuration item that requires licensing.

Tips

License types

Projects can and should negotiate licenses. There are many different kinds of software licenses. For example:

- *Enterprise-wide* licenses are negotiated for a whole organization's use of a component.
- *Per-seat* licenses are negotiated for each individual user.
- *Development-time* licenses are good for the engineering of a system that makes use of the license component, but not for the operational use.
- *End user or run-time* licenses are good for the operational use of the licensed component and are independent of any license required to make use of the component during engineering.

Opportunities

Capitalize on enterprise licensing opportunities.

- Alert the broader organization to the opportunity and/or the need for enterprise licensing.
- Look for existing enterprise licenses the project can use.

Contracts

A license agreement may be expressed as a contract, and a contract may be with a vendor as well as an integrator. It is important to understand both vehicles and to coordinate their use across the organization.

Willingness to negotiate

Willingness to negotiate varies between vendors, so be realistic about what to expect from each one and how much it is really in the component's best interest for them to make changes especially for the project.

- In one instance, a major office component vendor told a major defense contractor that the defense contractor was not "large enough" to command special treatment (i.e., the inclusion of special features in their component). On the other hand, one DoD program was able to get the attention of even this major vendor through the promise of several million licenses.

Negotiator skills

Successful license negotiation depends on the negotiator's knowledge and skills, but no one on the project staff may have them. Not only must the negotiator have the communication and people skills necessary for sound negotiation, but they must also know what drives the particular vendor and what kinds of problems can arise in the future, as well as the kind of relationship that the organization wants to build with the vendor.

Impact on architecture and costs

License structures have an impact on architecture and costs (not only the costs associated with the licenses themselves, but also the costs to manage them).

- In one example, the system was to make use of a highly distributed database management system, but one of the vendors had assumed a centralized architecture and priced the licenses accordingly. Because of this architectural conflict, the cost of that component was prohibitive for that system, quickly eliminating the component from consideration.

Non-standard provisions

Since license agreements may broadly describe the relationship with a vendor, incorporate non-standard provisions, such as vendor commitment to including modification into the next commercial component release and the kind and degree of integration support to be provided by the vendor.

Terms over time

The license terms negotiated may not hold over time. The price can change, even after the original negotiation. In some instances, a component (e.g., Netscape Navigator)® for which you originally had to pay may subsequently be offered for free.

Component splits

As a component evolves, the vendor may decide to split its features or functionality between two or more components. This will likely require a new license agreement (or, more likely, multiple new licenses), perhaps at a substantial increase in costs. The acquirer can be proactive and reduce the impact of such vendor decisions by including in the licensing agreements guarantees that such a component split will have no impact on the project (i.e., that the original license will serve for the new component as well), at least

® Netscape Navigator is registered in the U.S. Patent and Trademark Office.

for some agreed-upon period of time after the component split.

Transfer

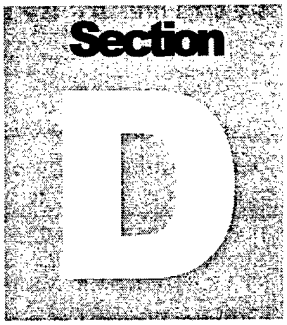
When negotiating licenses, keep in mind that it will in most cases be necessary in the future to transfer the license. Such a transfer might be to the broader organization or perhaps to a new contractor (should the project change contractors during the life of the system) or maintenance facility.

License "time bombs"

Vendors often protect the terms of their licenses by inserting in the software a software-license key or an expiration date after which the component will no longer function. Avoiding them may mean negotiating with the vendor to ensure license-management procedures that are adequate for protecting the vendor's interests.

Escrow accounts

The acquirer might consider the use of escrow accounts as risk mitigation against the vendor going out of business or ceasing support of a component critical to the project. In an escrow account, a neutral third party holds the designs, sources code, associated libraries, engineering environment, and pertinent documentation in trust for the contractor and the acquirer. The agreement will also detail the specifics of how these component materials will be stored (both media and format), how frequently they will be updated, where the media will be stored, the rights of specific individuals or organizations to audit or verify the content and condition of the media, etc. Under conditions that are spelled out in the escrow agreement, the acquirer (or the integration contractor, if they are party to the escrow) can take possession of the items held in escrow. But the project shouldn't enter into an escrow agreement without full awareness of what it would take to assume maintenance responsibility for that component, and an assessment that the project is fully prepared to so do both from a skill and a funding point of view.



Master Artifact List

The artifact list provides a summary of the models and other artifacts used throughout the process. EPIC uses artifacts from several sources.

- EPIC draws heavily on the RUP for technical and management artifacts associated with engineering processes. These artifacts are indicated with **(RUP artifact: found in <RUP artifact set name>)**, where <RUP artifact set name> refers to one of the categories of artifacts associated with the RUP, such as "Requirements Set." Readers should refer to [11] for guidelines and templates associated with these artifacts.

- A few of the listed artifacts are artifacts from the RUP that were renamed for EPIC. These artifacts are indicated with **(see RUP <RUP artifact name>: found in <RUP artifact set name>)**. Refer to [11] using the <RUP artifact name> for guidelines and templates.

- EPIC-unique artifacts, guidelines, and templates are found in Section C of this document and are indicated in this artifact list with **(EPIC artifact: found in Section C)**.

Readers who refer to the RUP sources should note that the RUP use of the terms *system* or *product* is equivalent to "solution" in EPIC.

The following shows the EPIC artifacts and their primary use. An alphabetized explanation of each artifact follows.

TO CHARACTERIZE END-USER BUSINESS PROCESSES AND STAKEHOLDER NEEDS

- Current Business Use-case Model
- Current Business Object Model
- Target Business Use-case Model
- Target Business Object Model
- Glossary
- Stakeholder Requests
- Solution Requirements Specification
- Use-case Model
- Use Cases
- Supplementary Specification

TO CHARACTERIZE THE MARKETPLACE AND OTHER SOURCES

- Market Segment Information
- Component Dossier (for each examined component)
- Component Screening Criteria and Rationale

TO CHARACTERIZE THE ARCHITECTURE AND DESIGN

- Solution Vision
- Architecture Document
- Design Model
- Executable Representation (s)
  - Implementation Model
- Test Set Artifacts (includes the Test Plan)
- Deployment Artifacts
  - End-user Support Materials (optional in first two phases)
  - Release Notes (required in Transition)
  - Training Materials (required in Transition)
  - Installation Artifacts (required in Transition)

DEVELOPMENT PLAN ELEMENTS TO CHARACTERIZE PROGRAMMATICS AND RISK

Management Process

- Project Plan
- Iteration Plans

Project Monitoring and Control

- Requirements Management Plan
- Schedule Control Plan
- Budget Control Plan
- Quality Control Plan
- Reporting Plan
- Measurement Plan

Risk Management

- Risk Management Plan

Technical Process Plans

- Development Case
- Infrastructure Plan
- Solution Acceptance Plan

Supporting Process Plans

- Configuration Management Plan
- Evaluation Plan
- Documentation Plan
- Quality Assurance Plan
- Problem Resolution Plan
- Vendor/Supplier Relationship Plan
- Process Improvement Plan

ADDITIONAL ARTIFACTS THAT CHARACTERIZE PROGRAMMATICS AND RISK

- Business Case (includes business context, success criteria, financial forecast)
- Business Process Change Management Plan
- Risk List
- Deployment Plan
- License Agreements
- Iteration Assessments (one/iteration)
- Review Record (one/phase)

Architecture Document

(see RUP artifact Software Architecture Document: found in Analysis & Design Set)

Provides a comprehensive architectural overview of the solution, using a number of different architectural views to depict different aspects of the solution (including component integration strategy). The Architecture Document is a key mechanism for capturing the architecturally significant project tradeoffs and decisions. It presents at least the views described below.

- The logical view shows the decomposition of the solution into a set of logical elements, i.e., classes, subsystems, packages, and collaborations.
- The process view maps the logical view elements to the processes and threads in the solution.
- The fielding view maps the processes to a set of nodes on which they execute.

Budget Control Plan

(RUP artifact: found in Project Management Set)

Describes the approach to be taken to monitor actual spending against the budget and what corrective action to take when required. The Budget Control Plan is in the Project Monitoring and Control section of the Development Plan.

Business Case

(RUP artifact: found in Project Management Set)

Transforms the vision into economic terms; answers the question as to whether the project is worth investing in. The Business Case summarizes each candidate solution, including financial estimates of the return on investment. The Business Case is created in the Inception Phase and updated with each iteration. Decisions to reject specific solutions are made based on the Business Case., which provides the following information on each candidate.

- solution context and scope
- technical approach (features, quality attributes, engineering tradeoffs and risk)
- management approach (project plan, acquisition strategy, schedule, schedule risk, and measures of success)
- evolutionary Appendices (financial forecast, etc.)

Business Object Model

(RUP artifact: found in Business Modeling Set)

Describes the realization of the business use cases. This model shows how the business use cases are performed in relation to interacting business workers and business entities. For EPIC, one model is constructed for the *current* state and another for the *target* state. The primary parts of the model include

- business workers–roles people play in an organization
- business entities–things an organization manages or produces

- relationships between entities and workers within the business

Business Process Change Management Plan

(EPIC artifact: found in Section C)

Describes the changes required to end-user business processes, organizational incentives, or organizational structure in the end-user organizations to use the solution; how those changes will be implemented; and how progress will be monitored. The plan includes any barriers to success and mitigation approaches.

Business Use-case Model

(RUP artifact: found in Business Modeling Set)

Captures the business's intended functions or processes as its customers and partners use (and interact with) the business. This model is an essential input to identify roles and deliverables of the target organization(s). For EPIC, one model is constructed for the *current* state and another for the *target* state. The primary parts of the model consists of

- Business actors–business users whose roles are external to the business, e.g., customers, vendors, partners
- Business use cases–business processes

Component Dossier

(EPIC artifact: found in Section C)

Provides an index that identifies and locates all of the information that represents the current understanding of a component. This information is produced and stored in different formats. Some of it (for example, contact information for the vendor) may be physically stored as part of the Component Dossier. Other information may be indexed by the Component Dossier but stored elsewhere. For example, the executable for a component may be represented in a tape library, component documentation may be at a network address, and reports produced by the vendor or the organization considering the component may be represented in a file cabinet. The purpose of the Component Dossier is to tie all of the divergent artifacts that represent a single instance of a component and use of that component together into a logical unit.

Component Screening Criteria and Rationale

(EPIC artifact: found in Section C)

Captures the identified criteria used to screen components. As the project proceeds, new or changed components will be introduced. These components should be screened based on current screening criteria. Early on, the criteria contain primarily basic component capabilities, vendor/supplier viability, and component scalability. As understanding grows regarding the stakeholder's needs and the components, the criteria evolve to include criteria used by previous **refines** to eliminate other components from consideration. The artifact also captures the rationale for removing any components from further consideration.

Configuration Management Plan

(RUP artifact: found in Configuration & Change Management Set)

Describes all configuration and change control management activities that are used throughout the life of the project. The plan captures the activities, when they are performed, who performs the activities, and the resources required to implement the plan.

Deployment Artifacts

(RUP artifacts: found in Deployment Set)

The set of artifacts necessary to support the Deployment Plan. Includes artifacts that identify changes, and known bugs, in a version of a build, or unit for fielding (Release Notes); the software and documented instructions required to install the solution (Installation Artifacts); and material that is used in training programs or courses to assist the end users with solution use, operation, and/or maintenance (Training Materials)

Deployment Plan

(RUP artifact: found in Deployment Set)

Describes the tasks required to install, test, and transition the selected solution into the user community. The plan covers the engineering of support materials such as training, management of acceptance testing, management of beta testing, full production packaging and distribution, resources required, rollout strategy to the user community, responsibilities of user sites, migration strategies in the user community, fielding schedule (and any sequencing requirements), and determining training requirements within the user community.

Design Model

(RUP artifact: found in Analysis & Design Set)

The major blueprint for the implementation of the solution. Captures the results of analysis and design into a single model. Analysis provides a rough sketch or generalization of the solution, omitting most of the detail. Design provides the details.

- The model consists of a set of collaborations of classes, packages, and subsystems that provide the behavior of the solution.
- The solution behavior is derived from the use-case model and non-functional requirements.

Development Case

(RUP artifact: found in Environment Set)

The Development Case specifies the tailored processes for a specific project. This includes which artifacts to use and how to use them, any modifications to the activities in the process, and assignment of responsibilities within the process.

Development Plan

(RUP artifact: found in Project Management Set)

A set of artifacts that describes how the project will be organized and managed. It references the entire set of major planning artifacts produced as part of the engineering and management activities. The Development Plan is used by the project manager and other project team members to understand what needs to be done, when, and by whom. The plan is a living artifact that is updated with each iteration.

Documentation Plan

(RUP artifact: found in Project Management Set)

Describes documents that will be created by the project for delivery to external stakeholders. One or more artifacts may make up a given document. Examples include development plans, requirements specifications, and master test plans. The Documentation Plan identifies the documents to be produced, schedule, scope of document, external recipients.

End-user Support Materials

(RUP artifact: found in Deployment Set)

Includes materials that assist the end user in learning, using, operating, and maintaining a solution. Examples of Support Materials include user guides, operational guides, maintenance guides, online demos, online help systems, and context-sensitive help systems.

End-user Support Material is typically required of any system that has a user interface. The initial planning of End-user Support Materials begins in the Inception and Elaboration Phases, as requirements and use cases evolve and the functionality of a solution is defined. End-user Support Materials are refined in the Construction Phase, in parallel with the development of the solution itself. Solutions with complex interfaces or with a lot of user interaction will require early versions of the user guide and also early prototypes of the interface. Embedded systems with little human interface will probably not require an early start on user documentation.

Evaluation Plan

(RUP artifact: found in Project Management Set)

Describes the project's plans for external system evaluation, and covers the techniques, criteria, metrics, and procedures used for evaluation with customers and end users. The Evaluation Plan includes walkthroughs, inspections, and reviews such as development or operational tests. The Evaluation Plan is part of the Software Development Plan. The Evaluation Plan is in addition to the Test Plan.

Executable Representation

An executable form that demonstrates the current agreed-upon state of the solution. In early iterations, the Executable Representation may be a mock-up of critical stakeholder needs. In later iterations, the Executable Representation is an evolutionary prototype that reflects the architecture and evolves to become the fielded solution. The Executable

Representation provides an ability to test the necessary infrastructure and any other systems with which the solution must interact. In addition, it provides the ability to prototype the end-user business processes inherent in the solution.

Glossary

(RUP artifact: found in Requirements Set)

Defines the important terms of the project. This establishes a common terminology for use across the project and organization. In addition, the Glossary captures the translation between the terminology used by the project and that used by the component's suppliers and the suppliers' other customers.

Implementation Model

(RUP artifact: found in Implementation Set)

Defines the collection of components, and the subsystems that contain them, that make up a solution. Components may be pre-existing, new custom code, data files, or component integration code.

Infrastructure Plan

(RUP artifact: referenced in the Development Plan: found in Project Management Set)

Describes the engineering and experimentation facilities needed to support the building, fielding, and ongoing support of the solution. The experimentation facilities are critical for the hands-on exploration of components and must be available from the Inception Phase until the solution is retired or replaced. The engineering and experimentation facilities include any hardware and software, such as computers and operating systems, on which engineering tools and components run, as well as any hardware and software used to interconnect computers and users.

Installation Artifacts

(RUP artifact: found in Deployment Set)

Describes how someone should install a solution. Installation Artifacts refer to the software and documented instructions required to install the release. Installation Artifacts are created in the Construction Phase and updated through the Transition Phase. Installation Artifacts include installation scripts, setup files, installation instructions, and so on.

Iteration Assessment

(RUP artifact: found in Project Management Set)

Captures the results of an iteration, including the degree to which the iteration's objectives were met, lessons learned, and recommended changes.

Iteration Plan

(RUP artifact: found in Project Management Set)

Fine-grained, detailed planning is conducted for each iteration using traditional planning techniques and tools. The iteration plan includes

- objective criteria for the success of the iteration
- concrete, measurable artifacts
- work breakdown structure
- duration and effort assigned to each activity
- schedule and assign work
- important dates
- plan for next iteration based on revised risk profile

License Agreement

(EPIC guidelines: found in Section C)

License agreements may take many forms. License negotiation is a set of activities for determining what the vendor offers with respect to terms, conditions, and costs for a given component for use by an organization over a particular period of time. Typically, a vendor has a set of usual licenses that it offers, but other ideas or additions may be negotiated.

Market Segment Information

(EPIC artifact: found in Section C)

Captures the broad characteristics of the market represented by a set of competing components that are under consideration for use in the system. This information includes vendors, suppliers, and buyers participating in the market, COTS components offered, processes automated, technologies represented, procurement strategies practiced, and competitive market forces. The focus of the artifact is on large-scale market dynamics rather than in-depth analysis of individual components.

Measurement Plan

(RUP artifact: found in Project Management Set)

Defines the measurement goals, associated metrics or indicators, what measures and indicators will be collected, and how they will be analyzed and reported.

Problem Resolution Plan

(RUP artifact: found in Project Management Set)

Describes the process used to report, analyze, and resolve problems that occur during the project. The Problem Resolution Plan is developed during the Inception Phase with scheduled updates based on the results of each Iteration Acceptance Review and Life-cycle

Milestone Review. Updates should also occur when changes to problem resolution procedures are identified through quality assurance reviews. The Problem Resolution Plan may physically be part of the Development Plan when the project environment is simple. If the project's interactions are complex (reviews, audits, assessments, and so on) with many stakeholders, it makes sense to have a separate artifact.

Process Improvement Plan

(RUP artifact: referenced in the Development Plan: found in Project Management Set)

Defines how the organization responsible for the engineering and support of the solution will work to improve its management and engineering processes. This includes how lessons learned are identified and analyzed, and how identified changes to the process are incorporated. Roles, responsibilities, schedules, and resources are defined.

Project Plan

(RUP artifact: found in Project Management Set)

The Project Plan is an integral part of the Development Plan and is updated as part of each iteration. The Project Plan captures

- the phases, number of iterations, major milestones, release points
- primary objectives and end dates for each iteration, if known
- resources required
- cost and schedule
- a work breakdown structure

Quality Assurance Plan

(RUP artifact: found in Project Management Set)

Captures how the project will ensure the quality of the solution, the artifacts used as part of the process, and the process itself.

Quality Control Plan

(RUP artifact: found in Project Management Set)

Describes the timing and methods to be used to control the quality of the project deliverables and how to take corrective action when required. The Quality Control Plan is in the Monitoring and Control section of the Development Plan.

Release Notes

(RUP artifact: found in Deployment Set)

Describes a release of a solution. Release Notes identify changes and known bugs in a version of a build, or deployment unit, that has been made available for either internal or external use.

Reporting Plan

(RUP artifact: found in Project Management Set)

Describes internal and external reports to be generated and the frequency and distribution of publication. The Reporting Plan is in the Monitoring and Control section of the Development Plan.

Requirements Management Plan

(RUP artifact: found in Requirements Set)

Captures the requirements-related information that will be used to track requirement status and the control mechanisms that will be used to collect, report, and control changes to the requirements.

Review Record

(RUP artifact: found in Project Management Set)

Captures the results of reviews of any artifact as part of the process; in particular, the Iteration Assessment associated with the conclusion of each phase.

Risk List

(RUP artifact: found in Project Management Set)

A list of known, open risks (technical and management) to the project with mitigation or contingency plans. The Risk List is a prioritized list and is integral to determining the objectives for each iteration.

Risk Management Plan

(RUP artifact: found in Project Management Set)

Describes how risks will be identified, analyzed, prioritized, monitored, and mitigated in the project.

Schedule Control Plan

(RUP artifact: found in Project Management Set)

Describes the approach to be taken to monitor progress against the planned schedule and how to take corrective action when required. The Schedule Control Plan is in the Monitoring and Control section of the Development Plan.

Solution Acceptance Plan

(see RUP artifact Product Acceptance Plan: found in Project Management Set)

Captures the criteria and the approach (e.g., tests) whereby the customer will evaluate the deliverable artifacts to determine if they satisfy the defined set of criteria. The plan also details the resources and responsibilities for the identified tasks.

Solution Requirements Specification

(see RUP artifact Software Requirements Specification: found in Requirements Set)

Captures the functional and non-functional requirements. This artifact contains the use cases and the Supplementary Specification.

Solution Vision

(see RUP artifact Vision: found in Requirements Set)

A vital document for a project, it summarizes both the problem and the solution at a high level of abstraction by capturing the main characteristics, major features, key stakeholders, needs, and key services to be provided by the solution. Provides the contractual basis for the high-priority requirements visible to the stakeholders.

Stakeholder Requests

(RUP artifact: found in Requirements Set)

Form a "wish list" or requests from the different stakeholders of what is expected or desired in the solution. Stakeholder requests are elicited, gathered, and analyzed to form the basis of the Solution Vision. The artifact includes

- brief description or explanation of the feature
- status (e.g., proposed, approved, incorporated, validated)
- estimated cost to implement in solution (e.g., types of resources, person-hours)
- priority (e.g., critical, important, ancillary)

Supplementary Specifications

(RUP artifact: found in Requirements Set)

Captures any functional and non-functional requirements for the solution that are not present in the Use-case Model. These are generally the non-functional requirements, but explicitly include any aspects of the broader organization's architecture that may constrain the design of the solution.

Test Set Artifacts

(RUP artifacts: found in Test Set)

A set of artifacts that plans and captures information associated with tests performed to assess the quality of the solution. The RUP set includes: Test Cases, Test Classes and Components, Test Evaluation Summary, Test Model, Test Package, Test Plan, Test Procedures, Test Results, Test Scripts, Test Subsystem, and Workload Analysis.

Training Materials

(RUP artifact: found in Deployment Set)

Include materials, depending on project requirements, used to teach users how to use, operate, or maintain a solution. Training Materials are initially created in the Elaboration Phase as the Requirements and Use Cases evolve and are refined in the Construction Phase.

The scope of Training Materials depends on the project's requirements (i.e., the needs of the affected end users). The materials include, as appropriate for classroom, web, or other computer-based learning media: overhead slides, student notes, teacher's instructions, example programs, databases, textbooks, and tutorials.

Use Cases

(RUP artifact: found in Requirements Set)

A use case defines a sequence of actions a solution performs that yields an observable result of value to a particular actor. The use case contains main, alternative, and exception flows of events. The functionality of a system is defined by different use cases, each of which represents a specific flow of events. The description of a use case defines what happens in the system when the use case is performed.

Use-Case Model

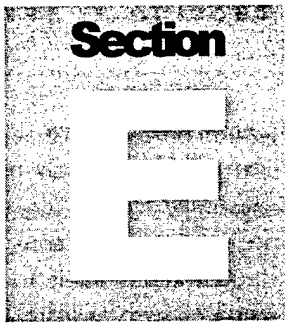
(RUP artifact: found in Requirements Set)

A use-case model is a model of the system's intended functions and its surroundings, and serves as a contract between the customer and the developers. Use cases serve as a unifying thread throughout system development. The most important purpose of a use-case model is to communicate the system's behavior to the customer or end user. The model contains use cases and actors. The users and any other system that may interact with the system are the actors. Because they represent system users, actors help delimit the system and give a clearer picture of what it is supposed to do. Use cases are developed based on the actors' needs. This ensures that the system will turn out to be what the users expected.

Vendor/Supplier Relationship Plan

(EPIC guidelines: found in Section C)

A vendor/supplier relationship is a cooperative exchange to explore current and future vendor/supplier and acquirer plans. Vendor/supplier relationships provide insights into current and future component releases and are a means for the acquirer to influence the vendor/supplier's plans and component directions. It is the primary means of partnering with vendors/suppliers whose components are important to the system. The Vendor/Supplier Relationship Plan describes the vendors/suppliers with whom to develop sound relationships, the nature of those relationships, who is responsible for what aspects of the relationships, and the effectiveness of the approaches.



Glossary

activity
A unit of work that a worker may be asked to perform. [11]

architecture
A high-level design that provides decisions made about the problem(s) that the component will solve, component descriptions, relationships between components, and dynamic operation description. [33]

A description of the essential elements of the system and the relationships between them. The elements include structure, behavior, usage, functionality, performance, resilience, reuse, comprehensibility, economic and technologic constraints and tradeoffs, and aesthetic issues. [11]

artifact
A piece of information that is produced, modified, or used by a process, defines an area of responsibility, and is subject to version control. An artifact can be a model, a model element, or a document. [11]

baseline
A reviewed and approved release of artifacts that constitutes an agreed-on basis for evolution or construction and that can be changed only through a formal procedure, such as change and configuration control. [11]

business process
The tasks, duties, or functions performed to support the objectives of an organization. [11]

commercial item
An item customarily used for nongovernmental purposes that has been or will be sold, leased, or licensed (or offered for sale, lease, or license) to the general public. An item that includes modifications customarily available in the commercial marketplace or minor modifications made to meet federal government requirements is still a commercial item. Services such as installation, maintenance, repair, and training procured for support of an item described above are considered commercial items if they are offered to the public under similar terms and conditions or sold competitively in substantial quantities based on established catalog or market prices. [34]

commercial off-the-shelf (COTS) component
A component that is (a) sold, leased, or licensed to the general public, (b) offered by a vendor trying to profit from it, (c) supported and evolved by the vendor, who retains the intellectual property rights, (d) available in multiple, identical copies; used without modification of the internals[20].

component
A nontrivial, nearly independent, and replaceable part of a system that fulfills a clear function in the context of a well-defined architecture. A component conforms to and provides the physical realization of a set of interfaces. [11]

In EPIC, the term "component" is used for one or more pre-existing hardware and software components from the commercial marketplace (i.e., COTS components), the legacy system (piece of the system being replaced), reuse libraries, or other reuse sources (e.g., freeware, shareware).

constraint
(1) A restriction, limit, or regulation. (2) A type of requirement that cannot be traded against other requirements. [33]

COTS-based system
Systems that are built out of various off-the-shelf parts. A COTS-based system can be one substantial COTS component that is tailored to provide significant system functionality or multiple components from a variety of sources integrated to collectively provide functionality. Sources can include items developed for, and in use by, other entity, legacy, custom, or "opportunity"-ware items, as well as COTS components. Many of these items may in turn include COTS components.

custom
Made to order. Uniquely built in direct response to a buyer's specifications.

customer
A purchaser or user of end components. [33]

[20] COTS-Based Systems for Program Managers (tutorial). Brownsword, L.; Oberndorf, P.; Sledge, C.; Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA. 1999.

One who buys goods or services. [35]

design
The part of the engineering process whose primary purpose is to decide *how* the system will be implemented. During design, strategic and tactical decisions are made to meet the required functional and quality requirements of a system. [11]

The set of decisions about a component that results in a common vision of what need it addresses, and how it addresses or satisfies that need. Typically, a design includes an operational concept (how users are expected or intended to use the component), components and their relationships, and sometimes decisions about the processes that will produce, field, and support it. [33]

end users
Those who will be using the system in the operational environment.

An individual or organization that uses, applies, or operates an end or enabling component. [33]

engineering
The integrated technical effort responsible for solution development that balances all factors associated with meeting system life-cycle requirements. [33]

evolutionary prototype
A prototype that evolves from one iteration to the next to become the final solution. Evolutionary prototypes tend to be designed rather formally and tested somewhat formally, even in early stages. Evolutionary prototypes are likely to use the infrastructure of the ultimate solution. [11]

exit criteria
Specific accomplishments or conditions that must be satisfactorily demonstrated before an effort can progress further in the current life-cycle phase or transition to the next phase. [33]

experimentation facility
Represents, as closely as practical, the operational environment in which hands-on experiments with components and/or a solution are conducted by engineers and end users.

feasible
A candidate solution is feasible if it describes a useful capability that can be integrated into the broader organization's architecture, in a reasonable period of time, at affordable cost, for acceptable risk, and based on available components.

fielding
Moving a version of the solution into some part of the end-user community.

harmonized
A solution is *harmonized* at a given level of abstraction when the candidate components can be assembled according to the evolving design to fulfill the agreed-upon requirements to support end-user business processes with acceptable cost, schedule, and risk.

Initial Operational Capability (IOC)
The Construction Phase ends with the Initial Operational Capability (IOC) anchor point. The IOC anchor point allows stakeholders to verify that a production-quality release of the solution is ready for fielding to at least a subset of the operational users as an initial fielding or beta test. [36]

integration
The engineering activity in which components are combined into an executable whole. [11]

The merger or combining of two or more elements (e.g., components, parts, or configuration items) into a functioning and higher level element with the functional and physical interfaces satisfied. [33]

Iteration
A distinct sequence of activities with a baselined plan and valuation criteria resulting in a release (internal or external). [11]

knowledge
Includes an increasingly detailed understanding of (a) the capabilities and limitations of candidate components, (b) the implications of the components on the requirements for the solution and the end user's business processes, as well as the planning necessary to implement any needed changes, and (c) the architectural alternatives and integration mechanisms that bind the components together.

life cycle
The scope of systems beginning with the identification of a perceived customer need, addressing engineering, test, manufacturing, operation, support, and training activities, and continuing through various upgrades or evaluation until the component disposal. [33]

Life-cycle Architecture (LCA)
The Elaboration Phase ends with the Life-cycle Architecture (LCA) anchor point milestone. Management verifies the basis for a sound commitment to development and evolution of a particular architecture that is shown to be feasible with respect to budget, schedule, requirements, operations concept, and business case and the elimination of all critical risk items. [36]

For EPIC at this point, all components have been selected and procured and any integration mechanisms necessary to incorporate the pre-existing components and any other components are validated.

Life-cycle Objective (LCO)
The Inception Phase ends with the Life-cycle Objectives (LCO) anchor point milestone. Management verifies the basis for a business commitment to proceed. [36]

In the RUP, LCO means that the requirements are settled sufficiently to form an architecture; in EPIC, LCO means one or more candidate solutions are identified that meet the solution's high-level objectives and have key stakeholder concurrence. The LCO anchor point reviews the phase exit criteria, determines that the phase objectives have been met, validates stakeholder concurrence on the scope of this solution, and seeks approval to examine the most viable candidate solutions in greater depth.

marketplace
The aggregation of buyers and sellers where goods are offered for sale, lease, or license.

model
A semantically closed abstraction of a system. In the RUP, a complete description of a system from a particular perspective–"complete"–meaning that you don't need additional information to understand the systems from that perspective. [11]

A simplified representation of some aspect of the real world. [33]

need
A user-related capability shortfall (such as those documented in a need statement, field deficiency report, or engineering change order), or an opportunity to satisfy a new market or capability requirement because of a new technology application or breakthrough, or to reduce costs. Needs may also relate to providing a desired service (e.g., system disposal). [33]

non-functional
Attributes of the solution that impose conditions on functional needs or requirements. Non-functionals address issues such as

- usability (e.g., human factors, aesthetics, consistency in the user interface, on-line help, user documentation, training materials)
- reliability (e.g., frequency/severity of failure, recoverability, predictability, accuracy, mean time between failure)
- performance (e.g., speed, efficiency, availability, accuracy, throughput, response time, recovery time, resource usage)
- supportability (e.g., testability, extensibility, adaptability, maintainability, compatibility, configurability, serviceability, installability, internationalization)
- physical characteristics (e.g., material used, shape, size, weight) [11]

phase
The time between two major project milestones during which a well-defined set of objectives is met, artifacts are completed, and decisions are made to move or not to move into the next phase. [11]

plan
A documented series of tasks required to meet an objective, typically including the associated schedule, budget, resources, organizational description, and work breakdown structure. [33]

procurement
The act of buying, leasing, or licensing components and/or services.

Pre-existing component
One or more pre-existing hardware and software components from the commercial marketplace (i.e., COTS components), the legacy system (piece of the system being replaced), reuse libraries, or other reuse sources (e.g., freeware, shareware).

production quality
A version of the solution that is built with sufficient quality for use in the end-user organization.

project
An engineering effort consisting of both technical and management activities for the purpose of engineering a system. [33]

prototype
An executable that is not necessarily subject to change management and configuration control. Behavioral prototypes focus on exploring specific behaviors of the system. Structural prototypes explore architectural or technological concerns. Exploratory prototypes are thrown away after they are finished and you have learned whatever you wanted from them. Evolutionary prototypes evolve to become the final system. [11]

A model (physical, electronic, digital, analytical, etc.) of a solution built or constructed for the purpose of (a) assessing the feasibility of a new or unfamiliar technology, (b) assessing or mitigating technical risk, (c) validating requirements, (d) demonstrating critical features, (e) qualifying a system, (f) qualifying a process, (g) characterizing performance or component features, or (h) elucidating physical principles. [33]

release
A subset of the end component that is the object of evaluation at a major milestone. [11]

requirement
A description of a condition or capability of a system that must be present; either derived directly from user needs or stated in a contract, standard, specification, or other formally imposed document. [11]

Something that governs that a component will have a given characteristic or achieve a given purpose, including what, how well, and under what conditions. [33]

reuse component
One or more pre-existing hardware and software components from the commercial marketplace (i.e., COTS components), the legacy system (piece of the system being replaced), reuse libraries, or other reuse sources (e.g., freeware, shareware).

risk
An ongoing or impending concern that has a significant probability of adversely affecting the success of major milestones. [11]

solution
A solution provides useful capability that can be fielded in a period of six to twelve months. A solution is the integrated assembly of the following:

- one or more pre-existing hardware and software components from the commercial marketplace (i.e., COTS components), the legacy system (piece of the system being replaced), reuse libraries, or other reuse sources (e.g., freeware, shareware)
- any required custom code (including wrappers and "glue")
- appropriate linkage to the broader organization's architecture with which the solution must interface
- any changes to the end user's business process necessary to match the processes provided in the components

spheres of influence
The term used to represent a set of information with common or related stakeholders, techniques for gathering and managing the information, and dynamic by which the information changes. EPIC focuses on four spheres of influence in forming solutions: 1) stakeholder needs and end-user business processes, 2) architecture and design, 3) the commercial marketplace and other sources, and 4) management of the project, planning and implementation for any needed changes to the end user's business process, and continuous refinement of the project cost, schedule, and risk. An emphasis on balance between the four spheres is critical to EPIC and must continue throughout the life of a project.

stakeholder
Any person or representative of an organization who has a stake–a vested interest–in the outcome of a project or whose opinion must be accommodated. A stakeholder can be an end user, a purchaser, a contractor, a developer, or a project manager. [11]

An individual or organization interested in the success of a component or system. Examples of stakeholders include customers, developers, engineers, managers, manufacturers, and end users, etc. [33]

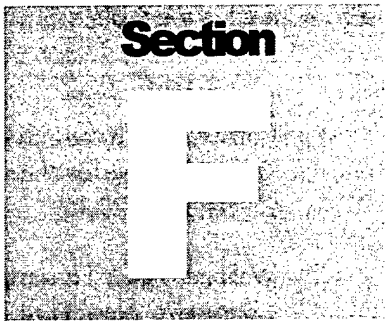
suppliers
Providers of components that are not COTS components.

tailoring
Using mechanisms that a supplier builds into a component to allow it to meet the specific needs of a particular system. Tailoring does not involve changes to the internal aspects of the component, such as source code. [7]

use case
A sequence of actions a system performs that yields an observable result of value to a particular actor. A use case contains all main, alternative, and exception flows of events related to producing the observable result of value. [11]

vendor
A commercial enterprise that produces and maintains COTS components. [7]

vision
The user's or customers' view of the component to be developed. [11]



References

1. Boehm, B. & Abts, C. "COTS Integration: Plug and Pray?" *IEEE Computer, 32,* 1 (January 1999):135-140.

2. Garlan, D.; Allen, R.; & Ockerbloom, J. "Architecture Mismatch: or Why It's Hard to Build System out of Existing Parts." *Proceedings of the International Conference on Software Engineering.* Seattle, WA, April 23-20, 1995. New York, NY: IEEE Computer Society Press, 1995.

3. Brownsword, L.; Carney, D.; & Oberndorf, P. "The Opportunities and Complexities of Applying COTS Components." *Crosstalk, 11,* 4 (April 1998). 4-6. <http://www.stsc.hill.af.mil/CrossTalk/1998/apr/applying.asp>.

4. Office of the Secretary of Defense, *Commercial Item Acquisition: Considerations and Lessons Learned.* <http://www.acq.osd.mil/ar/doc/cotsreport.PDF> (2000).

5. United States Air Force Science Advisory Board report on Ensuring Successful Implementation of Commercial Items in Air Force Systems, SAB-TR-99-03. <http://www.sab.hq.af.mil/archives/reports/1999/COTS/COTS_Report_Final_Public_Release .pdf>(2000).

6. Oberndorf, P.; Brownsword, L.; & Sledge, C. *An Activity Framework for COTS-Based Systems,* (CMU/SEI-2000-TR-010 ADA383836). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. <http://www.sei.cmu.edu/publications/documents/00.reports/00tr010.html>.

7. Meyers, B.C. & Oberndorf, P. *Managing Software Acquisition; Open Systems and COTS Products.* New York, NY: Addison-Wesley, SEI Series in Software Engineering, 2001.

8. Carney, D.; Hissam, S.; & Plakosh, D. "Complex COTS-based Software Systems: Practical Steps for their Maintenance." *Journal of Software Maintenance: Research and Practice, 12,* 6 (2000): 357-376.

9. Wallnau, K.C.; Hissam, S.A.; and Seacord, R.C. *Building Systems from Commercial Components.* New York, NY: Addison-Wesley, SEI Series in Software Engineering, 2002.

10. Kruchten, Phillippe. *The Rational Unified Process: An Introduction,* 2nd ed. New York, NY: Addison-Wesley Object Technology Series, March 2000.

11. Rational Unified Process, (software product) version 2000.02.10. Rational Software Corporation. <http://www.rational.com>.

12. Boehm, B. "A Spiral Model of Software Development and Enhancement." *IEEE Computer,* 21, 2 (February 1998): 61-72.

13. Clapp, J. & Engert, P. "Common Risks and Risk Mitigation Actions for a COTS-Based System." *Edge Perspectives* [An online MITRE newsletter]. <http://www.mitre.org/pubs/edge_perspectives/march_01/risks.html> (2001).

14. Garvey, P. & Lansdowne, Z. "Risk Matrix: An Approach for Identifying, Assessing, and Ranking Program Risks." *Air Force Journal of Logistics 25,* 1 (Spring 2001): 16-19.

15. Williams, R; Pandelios, G; & Behrens, S. *Software Risk Evaluation Method Description,* version 2.0 (CMU/SEI-99-TR-029 ADA 001008). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University. <http://www.sei.cmu.edu/publications/documents/99.reports/99tr029/99tr029abstract.html> (1999).

16. Park, R.; Goethert, W.; & Florac, W. *Goal-Driven Software Measurement–A Guidebook* (CMU/SEI-96-HB-002 ADA313946). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University. <http://www.sei.cmu.edu/publications/documents/96.reports/96.hb.002.html>(1996).

17. Lee, M.J. "Formal Modeling of the Win Win Requirements Negotiation System." Ph.D Dissertation, Computer Sciences Dept., University of Southern California, 1996.

18. Boehm. B.; Port, D.; Egyed, A.; & Madachy, R. "Using the Win Win Spiral Model: A Case Study." *IEEE Computer 13,* 7(July 1998): 33-44.

19. Fisher, R. & Ury, W. *Getting To Yes.* Boston, MA: Houghton-Mifflin, 1981.

20. Wellins, Richard S. & Murphy, Julie Schulz. "Reengineering: Plug into the Human Factor," *Training and Development 49*(1995): 33-37.

21. Kock, Ned F. *Process Improvement and Organizational Learning: The Role of Collaboration Technologies.* Hershey, PA: Idea Group, 1999.

22. Price Waterhouse Change Integration Team. *Better Change: Best Practices for Transforming Your Organization.* Chicago, IL: Irwin Professional Publishing, 1995.

23. Defaud, S. "Software Is the Mission." *CrossTalk* (October 1996): 24-28.

24. Przybylinski, S. "Software Technology Transition Tutorial," 13th *International Conference on Software Engineering,* Austin, TX, May 1991.

25. Brown, Mark G. *Baldrige Award Winning Quality.* White Plains, NY: Quality Resources; Milwaukee, WI: ASQC Quality Press, 1995.

26. Dunaway, D. *CMM Based Appraisal for Internal Process Improvement (CBA-IPI): Method Description* (CMU/SEI-96-TR-007 ADA307934). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996. <http://www.sei.cmu.edu/publications/documents/96.reports/96-tr-007/96-tr-007-abstract.html>.

27. Berk, Kevin J. & Hanrahan, Robert P. "Evaluating Tools and Environments Concept," 658-663.*Proceedings of the IEEE 1990 National Aerospace and Electronics Conference, NAECON 1990, 2* (May 1990) New York, NY: IEEE Computer Press, 1990.

28. Tornatzky, Louis G.; Fleisher, Mitchell with Chakrabarti, Alok K. [et al.] *The Process of Technology Innovation.* Lexington, MA: Lexington Books, 1990.

29. Martin, James & Odell, James J. *Object-Oriented Methods, A Foundation.* Englewood Cliffs, NJ: PTR Prentice Hall, 1995.

30. Champy, James. *Reengineering Management: The Mandate for New Leadership.* New York, NY: Harper Business, 1995.

31. Senge, Peter M. *The Fifth Discipline: The Art and Practice of the Learning Organization.* New York, NY: Doubleday/Currency, 1990.

32. Hamming, "One Man's View of Computer Science." *Turing Award Lectures: The First Twenty Years, 1966 to 1985.* New York, NY: ACM Press; Reading, MA.: Addison-Wesley, 1987.

33. *Interim Standard for Systems Engineering Capability Model.* EIA/IS 731.1, 1996.

34. *Federal Acquisition Regulations–Part 2.* Washington, DC: General Services Administration, 2001. <http://www.arnet.gov/far/current/pdf/FAR.book.pdf> (2001).

35. Webster, Noah. *Webster's II New College Dictionary.* Boston, MA: Houghton Mifflin Company, 1995.

36. Boehm, B. "Anchoring the Software Process." *IEEE Software* (July 1996): 73-82.

37. Implementation Management Associates. (developed under license from IMA, Copyright IMA 1989) *Managing Technological Change* (CMU/SEI-90-SR-20) Pittsburgh, PA.: Software Engineering Institute, Carnegie Mellon University, 1989.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	November 2002	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Evolutionary Process for Integrating COTS-based Systems (EPIC)	F19628-00-C-0003

6. AUTHOR(S)

Cecilia Albert, Lisa Brownsword

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2002-005

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116	ESC-TR-2002-005

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

Government and private organizations are escalating their use of commercial off-the-shelf (COTS) and other pre-existing components in critical business systems. Attempts to exploit these components have often failed with the use of traditional engineering approaches that involve defining requirements, formulating an architecture, and then searching for components that meet the specified requirements within the defined architecture. Here we describe an alternative approach using the Rational Unified Process (RUP), a risk-based, disciplined, spiral-engineering approach. The Evolutionary Process for Integrating COTS-based systems (EPIC)* redefines acquisition, management, and engineering practices to more effectively leverage the COTS marketplace and other sources of pre-existing components. This is accomplished through concurrent discovery and negotiation of diverse spheres of influence: user needs and business processes, applicable technology and components, the target architecture, and programmatic constraints. EPIC codifies these practices in a structured flow of key activities and artifacts. This document provides an overview of the EPIC framework along with its activities and artifacts.

14. SUBJECT TERMS	15. NUMBER OF PAGES
evolutionary process, integrating COTS-based systems, EPIC, ITSEP, Rational Unified Process, RUP, inception, elaboration, construction, transition, phase, iteration, COTS, spiral development, commercial.	278

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

* Previously known as Information Technology Solutions Evolution Process and Integrating Technology by a Structured Evolutionary Process (ITSEP).